AD-787 422

THREE-DIMENSIONAL FINITE-ELEMENT COMPUTER PROGRAM - USER'S GUIDE

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

PREPARED FOR

WATERVLIET ARSENAL

AUGUST 1974

DISTRIBUTED BY:

NTIS

National Technical Information Service

U. S. DEPARTMENT OF COMMERCE

BIBLIOGRAPHIC DATA SHEET

1. Report No.
VPI-E-74-19

2.

3. Recipient's Accession No.
AD 787422

4. Title and Subtitle
THREE-DIMENSIONAL FINITE-ELEMENT COMPUTER PROGRAM — USER'S GUIDE

5. Report Date
August 1974

6.

7. Author(s)
Jon R. Dana and Richard M. Barker

8. Performing Organization Rept. No. VPI-E-74-19

9. Performing Organization Name and Address
Engineering Science and Mechanics
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

10. Project/Task/Work Unit No.

11. Contract/Grant No.
DAA-F07-69-C-0444

12. Sponsoring Organization Name and Address
Department of Defense, U.S. Army
Contract No. DAA-F07-69-C-0444, with
Watervliet Arsenal, Watervliet, NY

13. Type of Report & Period Covered
Interim

14.

15. Supplementary Notes

16. Abstracts
The FORTRAN Listing and User's Guide for a Three-Dimensional Linear Thermal-Elastic Finite-Element Computer Program is presented. The program will determine three-dimensional displacement and stress distributions for laminated orthotropic composite materials.

A curved isoparametric element with 24 nodal points and 72 degrees-of-freedom is used to model the individual layers of a laminate. The nodal displacements are determined by minimizing the total potential energy of the system, at the element level, with a conjugate gradient iterative method.

The program is presently (1974) running on an IBM 370/158 computer at Virginia Polytechnic Institute and State University.

17. Key Words and Document Analysis. 17a. Descriptors

Composite Materials
Stress Distributions
Three Dimensional
Finite Element
Elastic Properties
Thermal Elastic
Computer Program
Structural Analysis
Conjugate Gradient

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement
Approved for public release; distribution unlimited

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page)
UNCLASSIFIED

21. No. of Pages
142

22. Price
5.75

i

VPI-E-74-19

THREE-DIMENSIONAL FINITE-ELEMENT COMPUTER PROGRAM—USER'S GUIDE

J. R. Dana* and R. M. Barker**

Department of Engineering Science & Mechanics

August 1974

Reporting Period: Interim

Prepared for:

Department of Defense, U. S. Army
Contract No. DAA-F07-69-C-0444, with
Watervliet Arsenal, Watervliet, New York

Approved for public release; distribution unlimited

*Department of Aeronautical Engineering
California Polytechnic State University
San Luis Obispo, California 93407

**Department of Civil Engineering
Virginia Polytechnic Institute & State University
Blacksburg, Virginia 24061

ABSTRACT

The FORTRAN Listing and User's Guide for a Three-Dimensional Linear Thermal-Elastic Finite-Element Computer Program is presented. The program will determine three-dimensional displacement and stress distributions for laminated orthotropic composite materials.

A curved isoparametric element with 24 nodal points and 72 degrees-of-freedom is used to model the individual layers of a laminate. The nodal displacements are determined by minimizing the total potential energy of the system, at the element level, with a conjugate gradient iterative method.

The program is presently (1974) running on an IBM 370/158 computer at Virginia Polytechnic Institute and State University.

NOMENCLATURE

E	Total potential energy
E_{11}	Modulus of elasticity parallel to the fibers
E_{22}	Modulus of elasticity transverse to the fibers
G_{12}, G_{13}, G_{23}	Shear modulus
m,n	Cos θ, sin θ respectively where θ is the angle between lamina and global axes
N	Number of global degrees of freedom
u, v, w	Displacements of a point in x, y, z directions
x, y, z	Global Cartesian coordinates
α	Magnitude of the correction vector
β	$\left\|\{r_{i+1}\}\right\|^2 / \left\|\{r_i\}\right\|^2$
θ	Fiber orientation angle
θ, r, z	Cylindrical coordinates for the global axes
λ_i	Eigenvalues
ξ, η, ζ	Local curvilinear coordinates
a_{ij}	Direction cosines for angle between lamina and global axes
C_{ijkl}	Stiffness constitutive relation for an anisotropic material in the local coordinate system
C'_{ijkl}	Stiffness constitutive relation for an anisotropic material in the global coordinate system
ϵ_{ij}	Strains in constitutive relations for an anisotropic material
ν_{ij}	Poisson's ratio relating normal strain in j-direction due to uniaxial normal stress in i-direction
{b}	Force vector
[D]	Elasticity matrix
[D]	A diagonal matrix

$[K]$	Global stiffness matrix
$\{N\}$	Shape functions
$[P]$	An orthogonal matrix
$\{p\}$	Direction of correction vector
$\{r\}$	Residue vector
$[T]$	Coordinate transformation matrix
$\{x\}$	Solution vector (displacements)
$\{x^*\}$	True solution vector (displacements)
$\{\varepsilon\}$	Error vector, $\{\varepsilon\} = \{x^*\} - \{x\}$
$\{\xi\}$	Change of variable vector, $\{\xi\} = [P]^T\{\varepsilon\}$

Subscripts:

1, 2, 3	Local system
x, y, z	Global system

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

The purpose of this computer program is to perform static, linear, thermal-elastic analyses of three-dimensional laminated composites. The basis for the analysis is a curved three-dimensional, isoparametric, 72 degree-of-freedom element with cubic interpolation functions in plan and a linear interpolation function through-the-thickness. This element can be used to model each layer of a laminated composite.

The primary deviation from the normal finite-element displacement formulation is that the global stiffness matrix is not formed. In this formulation only the unique element stiffness matrices are calculated. The nodal displacements are then determined by minimizing the total potential energy of the system at the element level with a conjugate gradient iterative method. The technique of not forming the global stiffness matrix greatly reduces the storage requiremencs if the number of unique elements is small. For example, problems of over 3000 degrees-of-freedom have been solved in core with less than 35,000 double precision words, including arrays and code. When the number of unique elements is greater than four, a direct access data file is used which increases the run time by about 60 percent.

The three nodal displacements (x, y and z) at each node obtained from the minimization technique are used in conjunction with the interpolation (shape) function to give the six stress components at each node. The stresses are calculated at the nodal points for each element.

The program and input data description that follow are intended to be used as a reference for a person with some knowledge of this

program. It is not written with sufficient detail to teach a person to use the program.

ANALYSIS

A. Three-Dimensional Isoparametric Lamina Element

The isoparametric element (Figure 1) used in this program was coded by Lin (7), and is similar to an element described by Ahmad, et al., (1) which was used to solve isotropic shell and plate problems. The development of the element stiffness matrix follows what has now become a standard procedure where the elastic properties related to the reference axes and the derivatives of the shape function related to the same axes through the Jacobian are used to form the strain energy density. The strain energy density is then numerically integrated (Gauss 4 x 4 x 2 rule) over the volume of the element to form the element stiffness matrix. Details of determining the necessary derivatives and forming the Jacobian matrix are given in the text by Zienkiewicz (10).

1. Interpolation function (shape function)

The triside nodes, top and bottom surfaces, are described by cubic interpolation functions while sections across the thickness are generated by straight lines. The relationship between the Cartesian coordinates (x, y and z) and the local normalized curvilinear coordinates (ξ, η, ζ) is given by

$$x = N_1x_1 + N_2x_2 + \dots + N_{24}x_{24} = \{N_i\}^T\{x_i\}$$

$$y = N_1y_1 + N_2y_2 + \dots + N_{24}y_{24} = \{N_i\}^T\{y_i\} \qquad (1)$$

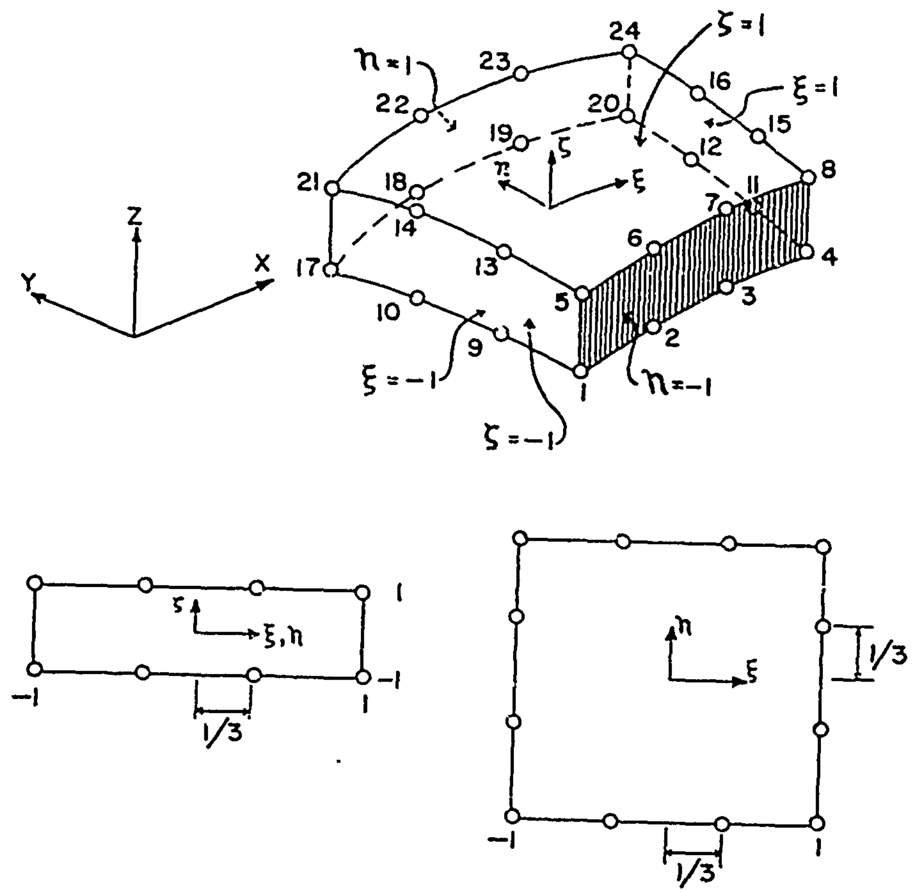

FIGURE 1: Three-Dimensional Isoparametric Lamina Element

$$z = N_1 z_1 + N_2 z_2 + \dots + N_{24} z_{24} = \{N_i\}^T \{z_i\}$$

where N_i are the isoparametric interpolation functions for the 24 nodal points. Introducing the notation

$$\xi_0 = \xi\xi_i, \quad \eta_0 = \eta\eta_i, \quad \zeta_0 = \zeta\zeta_i \tag{2}$$

the form for the interpolation functions becomes for the corner nodes with $\zeta_i = \pm 1$, $\eta_i = \pm 1$, and $\xi_i = \pm 1$

$$N_1 = \frac{1}{64}(1+\xi_0)(1+\eta_0)(1+\zeta_0)[9(\xi^2 + \eta^2) - 10]. \tag{3}$$

For nodes along the sides $\xi_i \pm 1$ with $\eta_i = \pm \frac{1}{3}$, $\zeta_i = \pm 1$

$$N_i = \frac{9}{64}(1+\xi_0)(1+9\eta_0)(1+\zeta_0)(1-\eta^2). \tag{4}$$

For nodes along the sides $\eta_i = \pm 1$ with $\xi_i = \pm \frac{1}{3}$, $\zeta_i = \pm 1$

$$N_i = \frac{9}{64}(1+9\xi_0)(1+\eta_0)(1+\zeta_0)(1-\xi^2). \tag{5}$$

The same functions are used to describe the displacement pattern (u, v, w) over the element in terms of the displacements (u_i, v_i, w_i) at the nodes, i.e.,

$$\begin{aligned} u &= \{N_i\}^T\{u_i\} \\ v &= \{N_i\}^T\{v_i\} \\ w &= \{N_i\}^T\{w_i\} \end{aligned} \tag{6}$$

2. Constitutive relation (material properties)

The constitutive relations used for the element are based on each lamina of the composite which is assumed to behave as an orthogonal anisotropic material. Therefore, the 21 elastic constants for a general anisotropic material are reduced to nine independent elastic constants which are given below, in matrix form, for the principal axes of elastic symmetry (1,2,3).

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & 0 & 0 & 0 \\ & D_{22} & D_{23} & 0 & 0 & 0 \\ & & D_{33} & 0 & 0 & 0 \\ & & & D_{44} & 0 & 0 \\ \text{Symmetry} & & & & D_{55} & 0 \\ & & & & & D_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} \quad (7)$$

where

$$D_{11} = \frac{1-\nu_{23}\,\nu_{32}}{F} E_{11}, \quad D_{22} = \frac{1-\nu_{13}\,\nu_{31}}{F} E_{22}, \quad D_{33} = \frac{1-\nu_{12}\,\nu_{21}}{F} E_{33},$$

$$D_{12} = \frac{\nu_{12}+\nu_{13}\,\nu_{32}}{F} E_{22}, \quad D_{13} = \frac{\nu_{13}+\nu_{12}\,\nu_{23}}{F} E_{33}, \quad D_{44} = G_{12},$$

$$D_{23} = \frac{\nu_{23}+\nu_{21}\,\nu_{13}}{F} E_{33}, \quad D_{55} = G_{13}, \quad D_{66} = G_{23}$$

and

$$F = 1-\nu_{12}\nu_{21}-\nu_{13}\nu_{31}-\nu_{23}\nu_{32}-\nu_{12}\nu_{23}\nu_{31}-\nu_{21}\nu_{13}\nu_{32}.$$

For an arbitrary orientation of the lamina, as shown in Figure 2, the principal axes (1,2,3) will not coincide with the reference axes (x, y, z) of the laminate; therefore, a rotational transformation must be performed. In general, the transformation takes the following tensor form:

$$C'_{ijkl} = a_{ir} a_{js} a_{kt} a_{lu} C_{rstu} \tag{8}$$

where

C'_{ijkl} and C_{rstu} are the components of a fourth order Cartesian tensor relating stresses and strains. The prime and unprimed components represent the reference axes and the principal axes, respectively, and

a_{mn} is a second order Cartesian tensor of direction cosines for a rotation about the z-axis.

Since C'_{ijkl} and C_{rstu} have 81 elements each and would be represented by fourth order arrays in FORTRAN, it is more convenient to perform the transformation in matrix form as shown below.

$$[D_x] = [T]^T [D_1] \ [T] \tag{9}$$

where

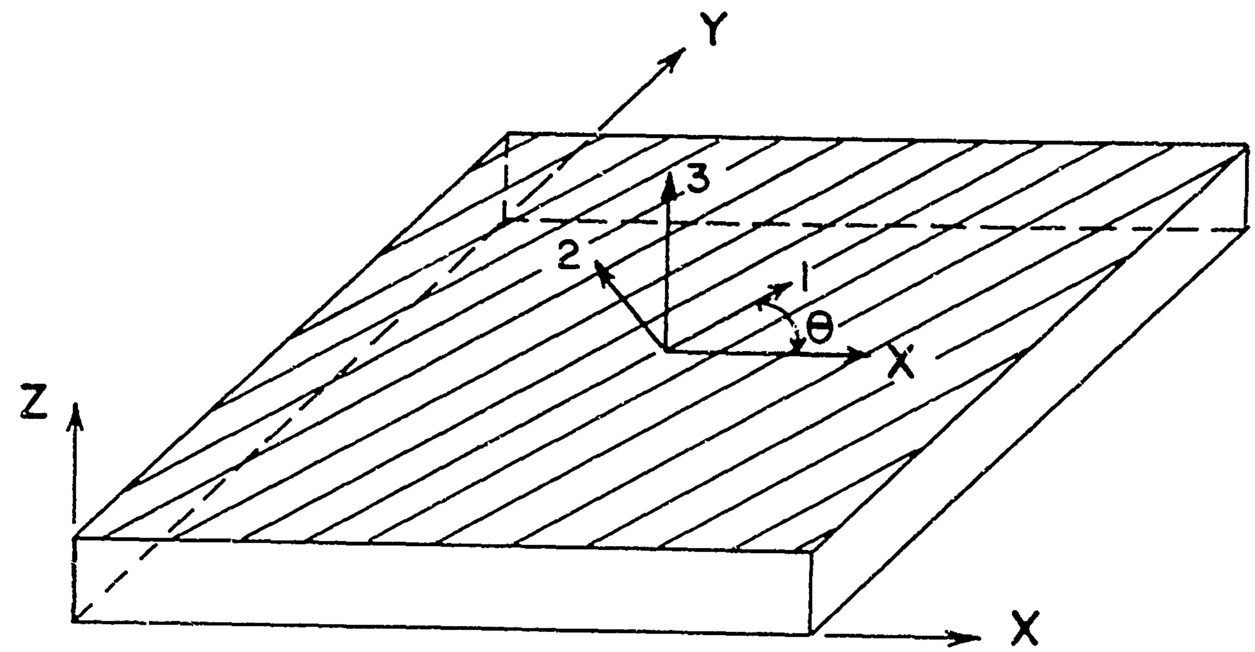

FIGURE 2: Fiber Orientation Within Lamina Element

$$[T] = \begin{bmatrix} m^2 & n^2 & 0 & -2nm & 0 & 0 \\ n^2 & m^2 & 0 & 2nm & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ nm & -nm & 0 & m^2-n^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & m & -n \\ 0 & 0 & 0 & 0 & n & m \end{bmatrix} \qquad (10)$$

and

$[D_x]$ and $[D_1]$ are the elastic matrices for the reference axes and principal axes, respectively.

Using references (2) and (8), it can be shown that the tensor transformation and the matrix transformation are equivalent for orthotropic materials. It should be noted that [T] and [D] are not Cartesian tensors; therefore, $[T]^{-1} \neq [T]^T$.

B. Conjugate Gradient Equation Solver

1. Description of the method

The equation solver used in this program is an adaptation of the conjugate gradient (CG) method originally presented by Hestenes and Stiefel (6) for linear systems in 1952. In more recent papers Fried (5) and Fox and Stanton (4) make direct reference to finite-element applications and indicate that the minimization process of the CG technique is equivalent to minimizing the total potential energy of the system. The method is an iterative process that will, apart from roundoff errors, converge to the exact solution in no more than

N iterations, where N is the order of the matrix.

The rate of convergence of the CG method is dependent on the eigenvalues of the global stiffness matrix (9); therefore, it is problem dependent, making it difficult to make a general comparison with other techniques. The dependence on the eigenvalues can be shown by considering the energy E which is to be minimized as

$$E = \frac{1}{2}\{x\}^T[K]\{x\}-\{x\}^T\{b\} \tag{11}$$

where [K] is the global stiffness matrix, {b} is the force vector, and {x} is the displacement vector which is to be selected to minimize the total potential energy. The energy E will be a minimum at the point

$$\{x\} = \{x^*\} \tag{12}$$

when

$$\frac{\partial E}{\partial x} = [K]\{x^*\} - \{b\} = 0. \tag{13}$$

At a particular step in the iteration process

$$\{x\} = \{x^*\}+\{\varepsilon\} \tag{14}$$

where the vector {ε} is the error in {x}. Putting equation (14) into equation (11) yields

$$E = \frac{1}{2}(\{x^*\}+\{\varepsilon\})^T[K](\{x^*\}+\{\varepsilon\})-(\{x^*\}+\{\varepsilon\})^T\{b\} \tag{15}$$

which after some manipulation and use of equation (13) becomes

$$E = \frac{1}{2}\{\varepsilon\}^T[K]\{\varepsilon\} - \frac{1}{2}\{x^*\}^T\{b\}. \qquad (16)$$

This can also be written as

$$E + \frac{1}{2}\{x^*\}^T\{b\} = \frac{1}{2}\{\varepsilon\}^T[K]\{\varepsilon\} = \frac{1}{2}k_{ij}\varepsilon_i\varepsilon_j = S \qquad (17)$$

where S is a hyperellipsoidal surface in variable $\{\varepsilon\}$, with center at $\{\varepsilon\}$ equal $\{0\}$. Since [K] is symmetric, there exists an orthogonal matrix [P] such that

$$[P]^T[K][P] = \lceil D \rfloor \qquad (18)$$

where $\lceil D \rfloor$ is a diagonal matrix containing the eigenvalues λ_i of [K]. Using the change of variable,

$$\{\xi\} = [P]^T\{\varepsilon\}. \qquad (19)$$

Equation (17) can be written as

$$S = \frac{1}{2}\{\varepsilon\}^T[K]\{\varepsilon\} = \frac{1}{2}\{\xi\}^T\lceil D \rfloor\{\xi\} = \frac{1}{2}\lambda_i\xi_i^{\,2}. \qquad (20)$$

The surface S described by equation (20) is shown for the two-dimensional case in Figure 3. The major and minor axes of the ellipse are proportional to the inverse of the square root of the eigenvalues of [K]. The vector {p} is normal to the surface of the ellipse and indicates the direction in which {x} will be corrected. It can be seen that, if λ_1 and λ_2 are similar in magnitude, the ellipse approaches a circle and {p} will be directed toward the

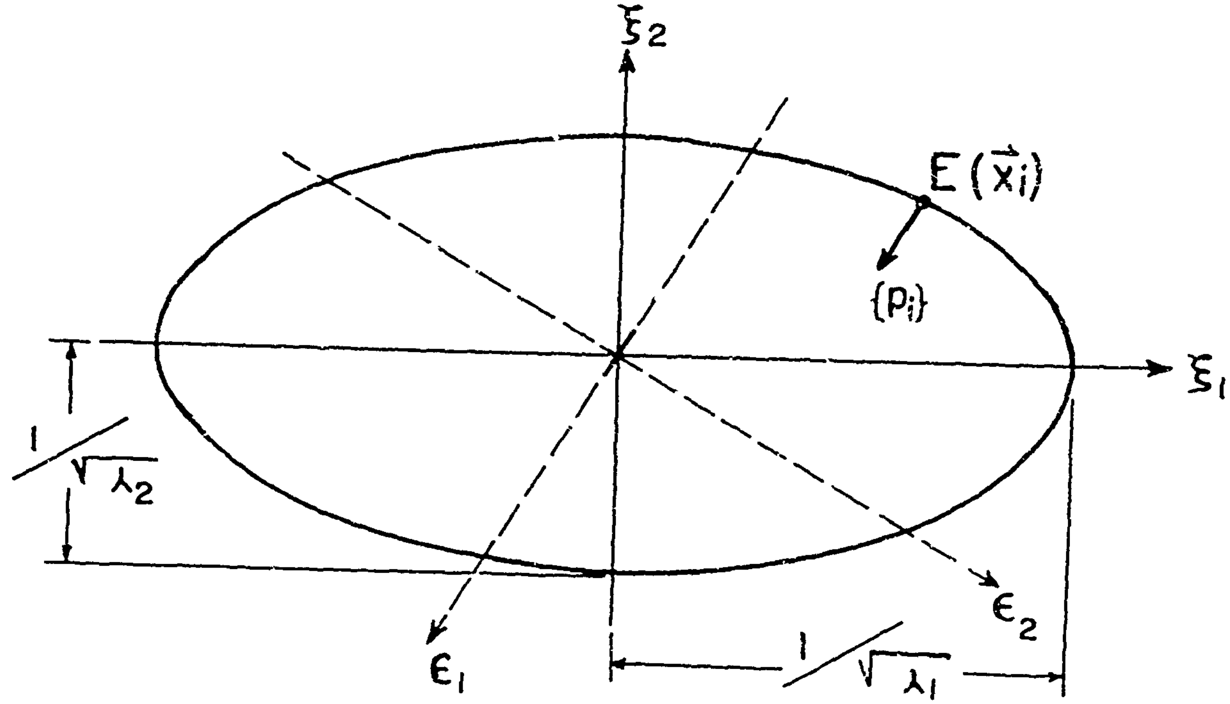

FIGURE 3: Two-Dimensional Representation of the Conjugate Gradient Procedure.

origin where ε_1 and ε_2 are zero. However, if the magnitudes of λ_1 and λ_2 are quite different, then {p} will not be in the direction of the origin and convergence to ε_1 and ε_2 in the neighborhood of zero will be slower.

The CG algorithm given by Hestenes and Stiefel is

$$\{p_o\} = \{r_o\} = \{b\} - [K]\{x_o\}$$

$$\alpha_i = \frac{|\{r_i\}|^2}{\{p_i\}^T[K]\{p_i\}}$$

$$\{x_{i+1}\} = \{x_i\} + \alpha_i\{p_i\}$$

$$i = i+1 \qquad (21)$$

$$\{r_{i+1}\} = \{r_i\} - \alpha_i[K]\{p_i\}$$

$$\beta_i = \frac{|\{r_{i+1}\}|^2}{|\{r_i\}|^2}$$

$$\{p_{i+1}\} = \{r_{i+1}\} + \beta_i\{p_i\}$$

where {r} is the residue vector, {p} is a vector representing the direction in which {x} is corrected, α is a scalar correction of the magnitude of {x}, and [K] is the global stiffness matrix.

The matrix [K] is shown in the algorithm but is not stored as an assembled global stiffness matrix in the computer. Instead, the matrix-vector products {Kx} and {Kp} are formed at the element level with {Kx} determined once at the beginning and with {Kp} formed for each iteration in the process.

2. Convergence Criteria

The equation solver uses two tests for convergence. First the residue vector must be less than unity. The second test is on strain energy; the change in strain energy normalized with strain energy must be less than a 'test value' specified by the user. Since this change in strain energy test is only a rate of convergence test, the final decision to accept a solution must be left to the user. References (3) and (5) can be used to estimate the accuracy of a solution for some classes of problems.

DESCRIPTION OF INPUT-OUTPUT DATA

The program consists of three or four FORTRAN job steps: (1) mesh generator (optional), (2) stiffness matrix formulation, (3) equation solver, and (4) stress calculations. Any one of the job steps can be run as a separate program provided the proper JCL is used. Input data for step (2) (Stiffness matrix formulation) can be from cards or passed from step (1) by card images on a sequential disk or tape unit. When data is passed from step (1) to step (2) by disk or tape, only one data card is required for step (2) to specify the disk or tape unit number. Data is passed from steps (2) to (3) and (3) to (4) by direct access data files only. One control data card each is required for job steps (3) and (4). Job step (3) can be restarted to break up jobs with long run times.

The program will handle four different types (called <u>classes</u>) of problem. The types are:

Class 1 - linear elastic with constant material properties within an element

Class 2 - linear elastic with material properties varying within an element

Class 3 - linear thermal elastic with constant material properties within an element

Class 4 - thermal elastic with temperature dependent material properties

A. Input Data for Step 2 (Stiffness matrix formulation)

1. <u>Input unit card</u> (I5) one card

Columns 1-5 unit number (specifies the unit which the input data will be read from, e.g., unit five will read data from cards)

2. Heading card[1] (10A8) one card

Columns 1-80 information to be printed with output

3. Control parameter card[1] (4I5, F10.2) one card

Columns 1-5 total number of nodal points

6-10 number of different materials

11-15 total number of elements

16-20 number of unique elements

21-30 initial temperature

4. Material data cards[1]

A different material must be specified if any of the nine orthotropic constants, the fiber orientation, or the three thermal expansion coefficients are changed. Two cards are necessary for each material if the problem class is 1, 2, or 3. If the problem is class 4 then the nine elastic orthotropic constants are expressed as a function temperature. From one to nine sets of temperature-dependent elastic constants can be specified for each material. A linear interpolation is used to determine material properties between temperature-specified sets of elastic constants, and the material properties are assumed to be constant above the highest specified temperature and below the lowest specified temperature. The temperature-dependent cards must be in ascending order of temperatures.

First card (2I5, F10.2, 3F10.8) one for each material

Columns 1-5 material number (in sequential order)

6-10 number of temperature cards for this material ('1' for class 1, 2, or 3)

11-20 fiber orientation in degrees

21-30 thermal expansion coefficient, α_{11}

[1]The word 'card' also implies card images on disk or tape

31-40 thermal expansion coefficient, α_{22}

41-50 thermal expansion coefficient, α_{33}

Subsequent cards (F5.0, 3F10.0, 3F5.2, 3F10.0) (One card for problem class 1, 2 or 3. And for problem class 4 one card for each temperature for which material properties are specified.)

Columns 1-5 temperature for material properties (can be left blank for class 1 and 2 problems)

Columns 6-15 modulus of elasticity, E_{11}, KSI

16-25 modulus of elasticity, E_{22}, KSI

26-35 modulus of elasticity, E_{33}, KSI

36-40 Poisson's ratio, ν_{12}

41-45 Poisson's ratio, ν_{13}

46-50 Poisson's ratio, ν_{23}

51-60 shear modulus, G_{12} KSI

61-70 shear modulus, G_{13} KSI

71-80 shear modulus, G_{23} KSI

5. Element data cards[1] (16I5) Two cards for each element. Figure 1 shows the element nodal numbers.

First card

Columns 1-5 element number (sequential)

6-10 global nodal number for element nodal number 1

11-15 global nodal number for element nodal number 2 (Global nodal numbers are put in fields of 5 columns for sequential element nodal numbers up to element nodal number 15 in columns 76-80.)

Second card

Columns 1-5 global nodal number for element nodal number 16 (Global nodal numbers are put in fields of 5 columns for sequential element nodal numbers up to element nodal number 24 in columns 41-45.)

46-50 material number

51-55 element type number

[1]The word 'card' also implies card images on disk or tape

(Each unique element is given a type number. The element types are numbered sequentially from one to the number of unique elements.)

56-60 class number (to specify type of thermal elastic problem)

'1' - elastic only, constant material properties within an element

'2' - elastic only, material properties can vary within an element

'3' - thermal elastic, material properties cannot vary with temperature within an element

'4' - thermal elastic, material properties can vary with temperature within an element

(Class 1 or 2 elements cannot be mixed with class 3 and 4 elements. Classes 1 and 2 can be mixed and classes 3 and 4 can be mixed.)

6. Nodal point cards[1] (I4,I4,I2, 6F10.6, F10.2) One card for each nodal point.

Columns 1-4 nodal point number (sequential)

5-8 material

(Only necessary if the material at this node is different from the material specified for the element. This nodal material will be ignored for elements of class 1, 3 or 4.)

9-10 boundary condition code

(There are eight possible combinations of force, F, and displacement, U, boundary conditions for the x, y, and z coordinates at each node.)

[1] The word 'card' implies card images on disk or tape

'0'	F_x	F_y	F_z
'1'	U_x	F_y	F_z
'2'	U_x	U_y	F_z
'3'	U_x	F_y	U_z
'4'	F_x	U_y	F_z
'5'	F_x	F_y	U_z
'6'	F_x	U_y	U_z
'7'	U_x	U_y	U_z

11-20 x - coordinate (global system)

21-30 y - coordinate (global system)

31-40 z - coordinate (global system)

41-50 x - force or displacement boundary condition

51-60 y - force or displacement boundary condition

61-70 z - force or displacement boundary condition

71-80 final nodal temperature
(can be left blank for class 1 and 2 problems)

B. Card Input for Step 3 (Equation Solver)

Two different equation solvers are available. The first is an in-core version which is recommended for problems with less than four unique elements. The second version iterates from direct access disk and is recommended for problems with five or more unique elements. Both versions use the same input data.

1. Parameter control card (2I5, 2F10.0, I5)

Columns 1-5 code number for initial guess of the displacement vector

'1' - The same initial guess for each degree of freedom. The value of the initial guess is specified in columns 11-20.

'2' - The initial guesses for the displacement vector are to be read in from cards in

hexadecimal, FORMAT (5Z16). The initial guesses will be multiplied by the number specified in columns 11-20.

'3' - The initial guesses for the displacement vector are to be read in from cards in hexadecimal, FORMAT (5Z16). The initial guesses in the z-direction only will be multiplied by the number specified in columns 11-20.

'4' - The initial guesses for the displacement vector are read in from a direct access data file created in a previous job.

6-10 maximum number of iterations for this run

11-20 For code number 1, this field contains the initial guess.

For code number 2, this field contains a multiplication factor for all the degrees of freedom (use '1.0' if the initial guesses are not to be modified.)

For code number 3, this field contains a multiplication factor for the z-direction displacements only.

For code number 4, this field is not used.

21-30 Convergence criterion factor (use .000001)

31-35 Print-punch control code for displacements

'0' - No printed or punched output

'1' - Printed output only — no punch

'2' Punched output only — no print

'3' - Both printed and punched output

2. Displacement vector cards (5Z16)

The displacement vector deck is put behind the Parameter control card for code number 2 and 3 only.

C. Card Input for Step 4

1. Logic control card (I5) one card

   Columns 1-5 code to control printed displacements and stresses

   '0' - Stresses printed in a rectangular coordinate system only

   '1' - Stresses printed in a cylindrical coordinate system only

   '2' - Displacements printed in a cylindrical coordinate system only

   '3' - Both displacements and stresses printed in a cylindrical coordinate system

   '4' - Both displacements and stresses printed in cylindrical coordinate system plus stresses in a rectangular coordinate system

D. Printed Output from Step 2

1. Problem parameters

2. Material properties

3. Local-to-global correlation matrix (element data)

   a. EL NO - Element number

   b. L _ _ - Lower _____ _____

   c. U _ _ - Upper _____ _____

   d. _ F _ - _____ Front _____

   e. _ M _ - _____ Middle _____

   f. _ B _ - _____ Back _____

   g. MT - Material number

   h. ET - Element type

   i. C - Class

4. Nodal point data

   a. NODE - Nodal number

   b. MATL - Material number

   c. CODE - Boundary condition code

d. X-COORD - x-coordinate in global system

e. Y-COORD - y-coordinate in global system

f. Z-COORD - z-coordinate in global system

g. X-DISPL/LOAD - Value of x boundary condition

h. Y-DISPL/LOAD - Value of y boundary condition

i. Z-DISPL/LOAD - Value of z boundary condition

E. Printed Output from Step 3

1. Convergence parameters

2. Displacements in rectangular coordinates

F. Printed Output from Step 4

1. Displacements in cylindrical coordinates

2. Stresses in cylindrical coordinates

3. Stresses in rectangular coordinates

ACKNOWLEDGMENTS

This research was supported by the Department of Defense, U. S. Army, Contract DAA F07-69-C-0444, with Watervliet Arsenal, Watervliet, N.Y. This financial support is gratefully acknowledged. The authors would also like to thank Susan Lee for typing this manuscript.

REFERENCES

1. Ahmad, S., Irons, B. M., and Zienkiewicz, O. C., "Analysis of Thick and Thin Shell Structures by Curved Finite Elements," International Journal of Numerical Method in Engineering, Vol. 2, 1970, pp. 419-451.

2. Ambartsumyan, S. A., Theory of Anisotropic Plates, Technomic Publishing Co., 1970.

3. Dana, Jon R., "Three Dimensional Finite Element Analysis of Thick Laminated Composites — Including Interlaminar and Boundary Effects Near Circular Holes," Ph.D. Dissertation, Virginia Polytechnic Institute & State University, August 1973.

4. Fox, R. L., and Stanton, E. L., "Developments in Structural Analysis by Direct Energy Minimization," AIAA Journal, Vol. 6, No. 6, June 1968, pp. 1036-1042.

5. Fried, I., "A Gradient Computational Procedure for the Solution of Large Problems Arising from the Finite Element Discretization Method," International Journal of Numerical Method in Engineering, Vol. 2, 1970, pp. 477-494.

6. Hestenes, M. R., and Stiefel, E., "Methods of Conjugate Gradients for Solving Linear Systems," Journal of Research of National Bureau of Standards, Vol. 49, No. 6, Research Paper 2379, 1952.

7. Lin, F. T., "The Finite Element Analysis of Laminated Composites," Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1971.

8. Musgrave, M. J. P., Crystal Acoustics, Holden-Day, San Francisco, 1970.

9. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill, New York, 1965.

10. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, 2nd Ed., McGraw-Hill, London, 1971.

APPENDIX A

FORTRAN Listing of Program

Appendix contains the FORTRAN listing for:

STEP 2 - Stiffness matrix

STEP 3 - Equation solver
(Iteration in-core version)

STEP 3 - Equation solver
(Iteration from disk version)

STEP 4 - Stress components

```
C     MAIN PROGRAM STEP 2
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                 *
C     *   STEP 2 PERFORMS FOUR FUNCTIONS                                *
C     *                                                                 *
C     *   1.     READS MESH AND BOUNDARY CONDITION DATA FROM CARDS OR   *
C     *          CARD IMAGES
C     *                                                                 *
C     *   2.     CALCULATES THE UNIQUE ELEMENT STIFFNESS MATRICES       *
C     *                                                                 *
C     *   3.     CALCULATES THERMAL EQUIVALENT LOADS                    *
C     *                                                                 *
C     *   4.     WRITES MESH DATA, UNIQUE ELEMENT STIFFNESS MATRICES AND *
C     *          FORCE AND DISPLACEMENT BOUNDARY CONDITION VECTORS ON A *
C     *          DIRECT ACCESS DATA SET                                 *
C     *                                                                 *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                 *
C     *   VARIABLE DEFINITIONS AND DIMENSIONS FOR STEP 2                *
C     *                                                                 *
C     *   ALFA1(NMTL) - - -  THERMAL EXPANSION COEFFICIENT 11           *
C     *                                                                 *
C     *   ALFA2(NMTL) - - -  THERMAL EXPANSION COEFFICIENT 22           *
C     *                                                                 *
C     *   ALFA3(NMTL) - - -  THERMAL EXPANSION COEFFICIENT 33           *
C     *                                                                 *
C     *   AMBTMP - - - - - - INITIAL TEMPERATURE                        *
C     *                                                                 *
C     *   DBC(NGNP) - - - -  DISPLACEMENT BOUNDARY CONDITIONS           *
C     *                                                                 *
C     *   E(NMTL,9,NTMP) - - MATERIAL PROPERTIES                        *
C     *                      (1ST SUBSCRIPT - NUMBER OF MATERIALS)      *
C     *                      (2ND SUBSCRIPT - NUMBER OF ELASTIC CONSTANTS*
C     *                      FOR AN ORTHOTROPIC MATERIAL)               *
C     *                      (3RD SUBSCRIPT - NUMBER OF TEMPERATURES THAT*
C     *                      MATERIAL PROPERTIES CAN BE SPECIFIED FOR   *
C     *                      EACH MATERIAL                              *
C     *                                                                 *
C     *   FBC(NGNP) - - - -  FORCE BOUNDARY CONDITIONS                  *
C     *                                                                 *
C     *   FIBORT(NMTL) - - - DIRECTION OF PRINCIPAL AXIS FOR EACH       *
C     *                      MATERIAL                                   *
C     *                                                                 *
C     *   GNMAT(NEL,72) - -  RELATES LOCAL AND GLOBAL DEGREES-OF-FREEDOM *
C     *                                                                 *
C     *   ICODE(NGNP) - - -  BOUNDARY CONDITION CODE (FORCE OR          *
C     *                      DISPLACEMENT)                              *
C     *                                                                 *
```

```
C     *   ICRD - - - - - - - UNIT NUMBER FOR CARD READER                    * MN2  520
C     *                                                                     * MN2  530
C     *   IUDA - - - - - - - UNIT NUMBER FOR DIRECT ACCESS FILE             * MN2  540
C     *                                                                     * MN2  550
C     *   IWRT - - - - - - - UNIT NUMBER FOR PRINTER                        * MN2  560
C     *                                                                     * MN2  570
C     *   IX(NEL,27) - - - - RELATES LOCAL AND GLOBAL NODAL POINTS          * MN2  580
C     *                                                                     * MN2  590
C     *   LDBC(NDBC) - - - - INDEX FOR DISPLACEMENT BOUNDARY CONDITIONS     * MN2  600
C     *                                                                     * MN2  610
C     *   MTLND(NGNP) - - -  MATERIAL AT EACH NODE                          * MN2  620
C     *                                                                     * MN2  630
C     *   NDBC - - - - - - - NUMBER OF DISPLACEMENT BOUNDARY CONDITIONS     * MN2  640
C     *                                                                     * MN2  650
C     *   NEL - - - - - - -  NUMBER OF ELEMENTS                             * MN2  660
C     *                                                                     * MN2  670
C     *   NGLDF - - - - - -  NUMBER OF DEGREES-OF-FREEDOM (GLOBAL SYSTEM)* MN2  680
C     *                                                                     * MN2  690
C     *   NGNP - - - - - - - NUMBER OF NODAL POINTS (GLOBAL SYSTEM)         * MN2  700
C     *                                                                     * MN2  710
C     *   NMTL - - - - - - - NUMBER OF MATERIALS                            * MN2  720
C     *                                                                     * MN2  730
C     *   NTMP(NMTL) - - - - NUMBER OF MATERIAL PROPERTIES SPECIFIED FOR * MN2  740
C     *                      EACH MATERIAL                                  * MN2  750
C     *                                                                     * MN2  760
C     *   NTYEL - - - - - -  NUMBER OF UNIQUE ELEMENTS                      * MN2  770
C     *                                                                     * MN2  780
C     *   S(2628) - - - - -  ELEMENT STIFFNESS MATRIX STORED AS ONE-        * MN2  790
C     *                      DIMENSIONAL ARRAY                              * MN2  800
C     *                                                                     * MN2  810
C     *   THC(NEL,72) - - -  THERMAL EQUIVALENT LOADS FOR EACH ELEMENT      * MN2  820
C     *                      (NEL CAN BE DIMENSIONED AS '1' FOR             * MN2  830
C     *                      NON-THERMAL PROBLEMS)                          * MN2  840
C     *                                                                     * MN2  850
C     *   TMPEL(NMTL,NTMP) - TEMPERATURES AT WHICH MATERIAL PROPERTIES      * MN2  860
C     *                      ARE SPECIFIED FOR EACH MATERIAL                * MN2  870
C     *                                                                     * MN2  880
C     *   TMPND(NGNP) - - -  FINAL NODAL POINT TEMPERATURES                 * MN2  890
C     *                                                                     * MN2  900
C     *   UX(NGNP) - - - - - MAGNITUDE OF FORCE OR DISPLACEMENT BOUNDARY * MN2  910
C     *                      CONDITIONS IN THE X-DIRECTION                  * MN2  920
C     *                                                                     * MN2  930
C     *   UY(NGNP) - - - - - MAGNITUDE OF FORCE OR DISPLACEMENT BOUNDARY * MN2  940
C     *                      CONDITIONS IN THE Y-DIRECTION                  * MN2  950
C     *                                                                     * MN2  960
C     *   UZ(NGNP) - - - - - MAGNITUDE OF FORCE OR DISPLACEMENT BOUNDARY * MN2  970
C     *                      CONDITIONS IN THE Z-DIRECTION                  * MN2  980
C     *                                                                     * MN2  990
C     *   X(NGNP) - - - - -  X-COORDINATE (GLOBAL SYSTEM)                   * MN2 1000
C     *                                                                     * MN2 1010
C     *   Y(NGNP) - - - - -  Y-COORDINATE (GLOBAL SYSTEM)                   * MN2 1020
```

```
C   *                                                                       *
C   *  Z(NGNP) - - - - -  Z-COORDINATE (GLOBAL SYSTEM)                       *
C   *                                                                       *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL*1  SW(12)
      INTEGER*2  IX, ICODE, GNMAT, MTLND, LDBC
      COMMON / GENL /  XINIT, EPS, AMBTMP,
     1       ICLASS, NEL, NGNP, NGLDF, NMTL, NTYEL, LIMIT, NM, NDBC
      COMMON  /NODELM/  X(1015), Y(1015), Z(1015), UX(1015), UY(1015),
     1  UZ(1015), TMPND(1015), FBC(3045), DBC(3045), TBC( 1,72),
     2   ICODE(1015), IX(144,27), GNMAT(144,72),MTLND(1015),LDBC(1015)
      COMMON  /MATL /   E(9,9,10),           FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9), TMPEL(9,10), NTMP(9)
      COMMON  /STIFEX/       S(2628)
      COMMON / INDX /   INEL, IGNP, ILNP,IMTL
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
      DEFINE FILE 3(55,6500,U,IDXDA)
      IUDA = 3
      CALL INPT12
      NGLDF = 3 * NGNP
C
C   CHANGE NODAL POINT NUMBERING SYSTEM TO DEGREE-OF-FREEDOM
C   NUMBERING SYSTEM
C
      DO 14 INEL=1,NEL
      DO 14 J=1,24
      GNMAT(INEL,3*J-2) = 3*IX(INEL,J)-2
      GNMAT(INEL,3*J-1) = 3*IX(INEL,J)-1
   14 GNMAT(INEL,3*J   ) = 3*IX(INEL,J)
C
C   ZERO THERMAL LOAD MATRIX
C
      IF(IX(1,27) .EQ. 1  .OR.  IX(1,27) .EQ. 2)  GO TO 5
      DO 15 I=1,NTYEL
      DO 15 J=1,72
   15 TBC(I,J) = 0D0
C
C   CALCULATE AND STORE UNIQUE STIFFNESS MATRICES
C
    5 IDXDA = 7
      DO 12 I     =1,NTYEL
      DO 20 INEL=1,NEL
      IF(IX(INEL,26)  .EQ.  I)  GO TO 6
   20 CONTINUE
    6 CALL ELSTIF
      WRITE(IUDA'IDXDA)    (S(J),J=1,2628)
   12 CONTINUE
C
C   FORM FORCE AND DISPLACEMENT VECTORS FROM BOUNDARY CONDITION DATA
```

```
C
      DO 1  I=1,NGLDF
      DBC(I) = 0.D0
    1 FBC(I)  =  0.D0
      CALL FBCDBC
C
C   COMBINE STATIC AND THERMAL LOADS
C
      IF(IX(1,27) .EQ. 1  .OR.  IX(1,27) .EQ. 2)  GO TO 7
      DO 17 INEL=1,NEL
      IX26 = IX(INEL,26)
      DO 17 I=1,72
   17 FBC(GNMAT(INEL,I)) = FBC(GNMAT(INEL,I)) + TBC(IX26,I)
    7 DO 46 J=1,12
   46 SW(J) = .FALSE.
C
C   WRITE PROBLEM DATA ON DISK TO BE PAST TO THE NEXT STEP
C
      WRITE(IUDA'1)      NEL, NGLDF,      NDBC, NTYEL, LIMIT, NGNP,NMTL,
     1       HED, IPAGE, AMBTMP
      WRITE(IUDA'2) ((NTMP(J), FIBORT(J), ALFA1(J), ALFA2(J), ALFA3(J)),
     1     ,TMPEL(J,I), (E(J,L,I),L=1,9), I=1,10), J=1,NMTL)
     2     , ((IX(I,J),J=1,27),I=1,NEL),
     3    (TMPND(J), MTLND(J),J=1,NGNP)
      WRITE(IUDA'3)   ,(X(J),J=1,NGNP), (Y(J),J=1,NGNP),(Z(J),J=1,NGNP)
      WRITE(IUDA'4)     (FBC(J),J=1,NGLDF)
     1    , SW, NOCNV,    (LDBC(J),J=1,NDBC)
      WRITE(IUDA'5)     (DBC(J),J=1,NGLDF)
     1                                      , (IX(J,26),J=1,NEL)
      WRITE(IUDA'6)    ((GNMAT(I,J),J=1,72),I=1,NEL)
      STOP
      END
```

```
      SUBROUTINE  INPT12
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE INPT12 READS MESH AND BOUNDARY CONDITION DATA FROM    *
C   *  CARDS OR CARD IMAGES AND PRINTS INPUT DATA                       *
C   *                                                                   *
C   *  THIS SUBROUTINE IS CALLED BY -                                   *
C   *        MAIN                                                       *
C   *                                                                   *
C   *  THIS SUBROUTINE CALLS -                                          *
C   *        TITLE                                                      *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2  IX, ICODE, GNMAT, MTLND, LOBC
      COMMON / GENL /  XINIT, EPS, AMBTMP,
     1      ICLASS, NEL, NGNP, NGLDF, NMTL, NTYEL, LIMIT, NM, NDBC
      COMMON  /NODELM/  X(1015), Y(1015), Z(1015), UX(1015), UY(1015),
     1  UZ(1015), TMPND(1015), FBC(3045), DBC(3045), TBC( 1,72),
     2   ICODE(1015), IX(144,27), GNMAT(144,72),MTLND(1015),LOBC(1015)
      COMMON  /MATL /   E(9,9,10),          FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9), TMPEL(9,10), NTMP(9)
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
1000 FORMAT(16I5)
1001 FORMAT(2I5,  F10.2, 3F10.8)
1002 FORMAT(F5.0, 3F10.0, 3F5.2, 3F10.0)
1003 FORMAT( I4, I4, I2, 6F10.6, F10.2)
1004 FORMAT(10A8)
1005 FORMAT( 4I5, F10.2)
2001 FORMAT(  //  '0 MATERIAL NO.      FIBER ORT.      THERMAL EXPAN. C
    1OEF. 11       THERMAL EXPAN COEF. 22      THERMAL EXPAN.COEF. 33')
2002 FORMAT  ('0', 4X,I5,13X,F5.1,10X, G16.7, 13X, G16.7, 13X, G16.7 /
    1        '0', 20X,'TEMP      E11        E22        E33        NU12
    2  NU13       NU23       G12        G13        G23')
2003 FORMAT(   19X, 4(F7.1,3X), 3(F7.2,3X), 3(F7.1,3X)  )
2004 FORMAT( '0 NODE MATL CODE    X-COORD          Y-COORD          Z-COORD
    1     X-DISPL/LOAD   Y-DISPL/LOAD   Z-DISPL/LOAD   TEMPERATURE'  )
2005 FORMAT (1X,2I5,I3,3X,G13.6, 6(2X, G13.6))
2006 FORMAT( '0EL NO LF1  LF2  LF3  LF4  UF1  UF2  UF3  UF4  LM1  LM2
    1LM3  LM4  UM1  UM2  UM3  UM4  LB1  LB2  LB3  LB4  UB1  UB2  UB3  U
    2B4 MT ET C'  )
2007 FORMAT (2X, I3, 24I5, 2I3, I2 )
2008 FORMAT('0DATA FOR THIS STEP IS READ IN ON UNIT',  I3)
2013 FORMAT('0PROBLEM CONSTANTS' /
    1    '       NUMBER OF NODAL POINTS',            T45,  I5/
    2    '       NUMBER OF MATERIALS',               T45,  I5/
    2    '       NUMBER OF ELEMENTS',                T45,  I5/
    3    '       NUMBER OF TYPES OF ELEMENTS',       T45,  I5/
    4  '       AMB. TEMPERATURE',           T37,E13.7)
```

```
C
C  READ IN PROBLEM DATA
C
      IPAGE = 1
      READ(5,1000)  INDAT
      READ(INDAT,1004) HED
      READ(INDAT,1005)  NGNP, NMTL, NEL, NTYEL, AMBTMP
C
C  PRINT PROBLEM DATA
C
      CALL TITLE
      WRITE(IWRT,2008)  INDAT
   15 WRITE(IWRT,2013)  NGNP, NMTL, NEL, NTYEL, AMBTMP
C
C  READ IN MATERIAL DATA
C
      DO 10 IMTL=1,NMTL
      READ(INDAT,1001)  MTLN, NTMP(IMTL), FIBORT(IMTL), ALFA1(IMTL),
     1       ALFA2(IMTL), ALFA3(IMTL)
      NTMP1 = NTMP(IMTL)
      DO 10 ITMP=1,NTMP1
   10 READ(INDAT,1002)   TMPEL(IMTL,ITMP), (E(IMTL,J,ITMP),J=1,9)
C
C  PRINT MATERIAL DATA
C
      CALL TITLE
      DO 50 IMTL=1,NMTL
      IF(LINE +.NTMP(IMTL)  .LT.  50 )  GO TO 1
      CALL TITLE
    1 WRITE(IWRT,2001)
      LINE = LINE + 3
      WRITE(IWRT,2002)  IMTL, FIBORT(IMTL),   ALFA1(IMTL), ALFA2(IMTL),
     1       ALFA3(IMTL)
      LINE = LINE+1
      NTMP1 = NTMP(IMTL)
      DO 50 ITMP=1,NTMP1
      LINE = LINE+1
   50 WRITE(IWRT,2003)   TMPEL(IMTL,ITMP),  (E(IMTL,J,ITMP),J=1,9)
C
C  READ IN ELEMENT DATA
C
      DO 30 INEL=1,NEL
   30 READ(INDAT ,1000)    M, (IX(M,J),J=1,27)
C
C  PRINT ELEMENT DATA
C
      CALL TITLE
      WRITE(IWRT,2006)
      DO 70 INEL=1,NEL
      IF(LINE .LT. 45)  GO TO 3
      CALL TITLE
```

```
      WRITE(IWRT, 2006)
    3 LINE=LINE+1
   70 WRITE(IWRT,2007)  INEL,  (IX(INEL,J),J=1,27)
C
C  READ IN NODAL POINT DATA
C
      DO 40 IGNP=1,NGNP
   40 READ(INDAT,1003)  M, MTLND(M), ICODE(M), X(M), Y(M), Z(M),
     1     UX(M), UY(M), UZ(M), TMPND(M)   .
C
C  PRINT NODAL POINT DATA
C
      CALL TITLE
      WRITE(IWRT,2004)
      DO 60 IGNP=1,NGNP
      IF(LINE  .LT.  45)  GO TO 2
      CALL TITLE
      WRITE(IWRT,2004)
    2 LINE = LINE+1
      WRITE(IWRT,2005)  IGNP, MTLND(IGNP), ICODE(IGNP), X(IGNP), Y(IGNP)
     1     , Z(IGNP), UX(IGNP), UY(IGNP), UZ(IGNP), TMPND(IGNP)
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TITLE
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE TITLE PRINTS THE HEADING ON EACH PRINTED PAGE          *
C   *                                                                   *
C   *  THIS SUBROUTINE IS CALLED BY -                                   *
C   *        INPT12                                                     *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
  100 FORMAT (1H1,'FEM 72-DOF GENERAL HEXAHEDRONS THERMO-ELASTIC, VARYIN
     1G MATERIAL PROPERTIES, DANA', 9X, 'PAGE', I3)
  101 FORMAT (1H0,10A8 )
      LIST = 6
      IWRT = 6
      WRITE (LIST,100) IPAGE
      WRITE (LIST,101) HED
      IPAGE= IPAGE +1
      LINE = 0
      RETURN
      END
```

```
      SUBROUTINE ELSTIF
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE ELSTIF CALCULATES THE ELEMENT STIFFNESS MATRICES.     *
C   *  ONLY THE UPPER SYMMETRIC PORTION IS FORMED.                      *
C   *  EACH MATRIX IS STORED AS A ONE-DIMENSIONAL ARRAY.                *
C   *                                                                   *
C   *  THIS SUBROUTINE IS CALLED BY -                                   *
C   *        MAIN                                                       *
C   *                                                                   *
C   *  THIS SUBROUTINE CALLS -                                          *
C   *        ELAS                                                       *
C   *        BMAT                                                       *
C   *        SHPFNT                                                     *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2  IX, ICODE, GNMAT, MTLND, LDBC
      COMMON / GENL /  XINIT, EPS, AMBTMP,
     1       ICLASS, NEL, NGNP, NGLDF, NMTL, NTYEL, LIMIT, NM, NDBC
      COMMON  /NODELM/  X(1015), Y(1015), Z(1015), UX(1015), UY(1015),
     1  UZ(1015), TMPND(1015), FBC(3045), DBC(3045), TBC( 1,72),
     2    ICODE(1015), IX(144,27), GNMAT(144,72),MTLND(1015),LDBC(1015)
      COMMON  /MATL /    E(9,9,10),            FIBORT(9), ALFA1(9),
     1    ALFA2(9), ALFA3(9), THPLL(9,10), NTMP(9)
      COMMON  /STIFEX/       S(2628)
      COMMON  / STIFIN /  D(6,6,24), XNVCT(24), ETM(9), BA(6,72),
     1     C(3,24), XYZ(24,3), ALFTMP(6), XSIK, ETAJ, ZTAI
      COMMON / INDX /    INEL, IGNP, ILNP,IMTL
      DIMENSION   DSHP(6,6), BTD(72,6), BDB(2628)
      DIMENSION H(32),XSI(4),ETA(4),ZTA(2),W(4),WW(2)
C
C FORM ELEMENT STIFFNESS MATRIX BY NUMERICAL INTEGRATION WITH GAUSS QUAD
C -RATURE FORMULAR, S(72X72) = H*(BT*D*B)*DETJ AT EACH G.P., 4X4X2 RULE
C SET GAUSS PT.S FOR GENERAL HEXAHEDRON IN NUM. INT. WITH 4X4X2 RULE
C
      XSI(4) = .86113631159405300
      XSI(3) = .33998104358485600
      XSI(2) = -XSI(3)
      XSI(1) = -XSI(4)
      ZTA(2) = .57735026918962600
      ZTA(1) = -ZTA(2)
      DO 10 I=1,4
   10 ETA(I) = XSI(I)
C
C FORM WEIGHING COEFFICIENTS H(32) AT GAUSS POINT
C
      W(1)=.34785484513745400
      W(2)=.65214515486254600
```

```
      W(3) = W(2)
      W(4) = W(1)
      WW(1) = 1.D0
      WW(2) = 1.D0
      DO 15 I=1,4
      DO 15 J=1,4
      DO 15 K=1,2
      M = K + 2*(J - 1) + 8*(I - 1)
   15 H(M) = W(I)*W(J)*WW(K)
C
C FORM NODAL PT. COORD.S MATRIX XYZ AND C FOR J(3X3) = J(3X24)*XYZ(24X3)
C
      DO 20 I=1,24
      L = IX(INEL,I)
      XYZ(I,1) = X(L)
      XYZ(I,2) = Y(L)
   20 XYZ(I,3) = Z(L)
C
C   CALCULATE ELASTIC PROPERTIES
C
      CALL ELAS
      DO 25 NL=1,2628
   25 S(NL) = 0.D0
      DO 100 K=1,4
      DO 100 J=1,4
      DO 100 I=1,2
      M = I + 2*(J - 1) + 8*(K - 1)
      XSIK = XSI(K)
      ETAJ = ETA(J)
      ZTAI = ZTA(I)
C
C   FORM B MATRIX
C
      CALL BMAT(I,J,K,DETJ)
      IF(IX(INEL,27) .EQ. 1 )  GO TO 1
      CALL SHPFNT
      IF( IX(INEL,27)  .EQ.  3 ) GO TO 1
C
C   FORM 3-D ELASTIC MATERIAL PROPERTIES ARRAY (USED ONLY IF ELASTIC
C   PROPERTIES VARY WITHIN AN ELELMENT)
C
      DO 110 N=1,6
      DO 110 L=1,6
      DSHP(L,N)  =  0D0
      DO 110 ILNP=1,24
  110 DSHP(L,N) = DSHP(L,N)  +  D(L,N,ILNP) * XNVCT(ILNP)
      GO TO 4
    1 DO 120 N=1,6
      DO 120 L=1,6
  120 DSHP(L,N) = D(L,N,1)
    4 CONTINUE
```

```
      DO 80 N=1,72
      DO 80 L=1,6
      BTD(N,L) = 0.D0
      DO 80 NN=1,6
   80 BTD(N,L) = BTD(N,L) + BA(NN,N)*DSHP(NN,L)
      IF( IX(INEL,27)  .EQ.  1 .OR.  IX(INEL,27) .EQ. 2) GO TO 5
C
C     CALCULATE THERMAL-EQUIVALENT LOAD VECTOR FOR EACH ELEMENT (THERMAL
C     PROBLEM ONLY)
C
      TMP = 0D0
      DO 130 ILNP=1,24
  130 TMP = TMP + TMPND(IX(INEL,ILNP))* XNVCT(ILNP)
      TMP = TMP - AMBTMP
      IX26 = IX(INEL,26)
      DO 140 L=1,72
      DO 140 N=1,6
  140 TBC(IX26,L) = TBC(IX26,L) + TMP*ALFTMP(N)*BTD(L,N)*H(M)*DETJ
    5 NL = 0
C
C     FORM TRIPLE MATRIX PRODUCT
C
      DO 90 N=1,72
      DO 90 L=N,72
      NL = NL+ 1
      BDB(NL) = 0.D0
      DO 90 NN=1,6
   90 BDB(NL) = BDB(NL) + BTD(N,NN)*BA(NN,L)
      NL=0
      DO 100 N=1,72
      DO 100 L=N,72
      NL = NL + 1
  100 S(NL) = S(NL) + H(M) * DETJ*BDB(NL)
      RETURN
      END
```

```
      SUBROUTINE ELAS
C
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    *                                                                   *
C    *  SUBROUTINE ELAS, IN CONJUCTION WITH DMAT, CALCULATES THE         *
C    *  ELASTIC MATRIX FOR EACH ELEMENT                                  *
C    *                                                                   *
C    *  THIS SUBROUTINE IS CALLED BY -                                   *
C    *        ELSTIF                                                     *
C    *                                                                   *
C    *  THIS SUBROUTINE CALLS -                                          *
C    *        DMAT                                                       *
C    *                                                                   *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2  IX, ICODE, GNMAT, MTLND, LDBC
      COMMON / GENL /  XINIT, EPS, AMBTMP,
     1        ICLASS, NEL, NGNP, NGLDF, NMTL, NTYEL, LIMIT, NM, NDBC
      COMMON  /NODELM/  X(1015), Y(1015), Z(1015), UX(1015), UY(1015),
     1  UZ(1015), TMPND(1015), FBC(3045), DBC(3045), TBC( 1,72),
     2    ICODE(1015), IX(144,27), GNMAT(144,72),MTLND(1015),LDBC(1015)
      COMMON  /MATL /   E(9,9,10),          FIBORT(9), ALFA1(9),
     1    ALFA2(9), ALFA3(9), TMPEL(9,10), NTMP(9)
      COMMON / INDX /   INEL, IGNP, ILNP,IMTL
      COMMON   / STIFIN /  D(6,6,24), XNVCT(24), ETM(9), BA(6,72),
     1     C(3,24), XYZ(24,3), ALFTMP(6), XSIK, ETAJ, ZTAI
      IF( IX(INEL,27) .EQ. 1  .OR.  IX(INEL,27) .EQ. 3)  GO TO 31
      IF( IX(INEL,27) .EQ. 2 )  GO TO 2
      NTMP1 = NTMP(IX(INEL,25))
      IF( NTMP1 .EQ. 1 )  GO TO 31
      DO 10 ILNP=1,24
      IMTL = MTLND(IX(INEL,ILNP))
      IF(IMTL .EQ. 0 )  IMTL = IX(INEL,25)
      IGNT = IX(INEL,ILNP)
      IF(TMPND(IGNT) .LT. TMPEL(IMTL,1) )  GO TO 5
      IF( TMPND(IGNT) .GE. TMPEL(IMTL,NTMP(IMTL))  )  GO TO 6
      NTMPM1 = NTMP(IMTL) - 1
      DO 20 II=1,NTMPM1
      IF( TMPND(IGNT).GT. TMPEL(IMTL,II)  .AND.  TMPND(IGNT).LE.
     1        TMPEL(IMTL,II+1) ) GO TO 4
   20 CONTINUE
    5 DO 30 I=1,9
   30 ETM(I) = E(IMTL,I,1)
      GO TO 1
    6 DO 40 I=1,9
   40 ETM(I) = E(IMTL,I,NTMP(IMTL))
      GO TO 1
    4 DIFTP1 = TMPEL(IMTL,II+1) - TMPEL(IMTL,II)
      DIFTP2 = TMPND(IGNT) - TMPEL(IMTL,II)
      RATDIF = DIFTP2 / DIFTP1
```

```
      DO 50 I=1,9
   50 ETM(I) = E(IMTL,I,II) + RATDIF * (E(IMTL,I,II+1) - E(IMTL,I,II))
      CALL DMAT
   10 CONTINUE
      RETURN
   31 IMTL = IX(INEL,25)
      ILNP = 1
      DO 60 I=1,9
   60 ETM(I) = E(IMTL,I,1)
      CALL DMAT
      RETURN
    2 DO 70 ILNP=1,24
      IMTL = MTLND(IX(INEL,ILNP))
      IF(IMTL .EQ. 0 )  IMTL = IX(INEL,25)
      DO 80 I=1,9
   80 ETM(I) = E(IMTL,I,1)
   70 CALL DMAT
      RETURN
      END
```

```
      SUBROUTINE  BMAT(I,J,K,DETJ)
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                               *
C   *  SUBROUTINE BMAT FORMS THE B MATRIX WHICH IS USED IN CONJUCTION *
C   *  WITH THE ELASTIC MATRIX TO FORM THE STIFFNESS MATRIX          *
C   *                                                               *
C   *  THIS SUBROUTINE IS CALLED BY -                               *
C   *        ELSTIF                                                 *
C   *                                                               *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2  IX, ICODE, GNMAT, MTLND, LDBC
      COMMON / GENL /  XINIT, EPS, AMBTMP,
     1        ICLASS, NEL, NGNP, NGLDF, NMTL, NTYEL, LIMIT, NM, NDBC
      COMMON  /NODELM/  X(1015), Y(1015), Z(1015), UX(1015), UY(1015),
     1  UZ(1015), TMPND(1015), FBC(3045), DBC(3045), TBC( 1,72),
     2    ICODE(1015), IX(144,27), GNMAT(144,72),MTLND(1015),LDBC(1015)
      COMMON  /MATL /   E(9,9,10),          FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9), TMPEL(9,10), NTMP(9)
      COMMON  / STIFIN /  D(6,6,24), XNVCT(24), ETM(9), BA(6,72),
     1     C(3,24), XYZ(24,3), ALFTMP(6), XSIK, ETAJ, ZTAI
      COMMON / INDX /   INEL, IGNP, ILNP,IMTL
      DIMENSION   DJ(3,3), DJI(3,3),XSI(4), ETA(4), ZTA(2)
      XSI(K) = XSIK
      ETA(J) = ETAJ
      ZTA(I) = ZTAI
C
C   FORM C MATRIX
C
      C(1,1) = (1.D0 - ETA(J))*(1.D0 - ZTA(I))*(10.D0 + 18.D0*XSI(K) -
     1           27.D0*XSI(K)**2 - 9.D0*ETA(J)**2)
      C(1,2) = (1.D0 - ETA(J))*(1.D0 - ZTA(I))*(81.D0*XSI(K)**2
     1           -18.D0*XSI(K) - 27.D0)
      C(1,3) = (1.D0 - ETA(J))*(1.D0 - ZTA(I))*(27.D0 - 18.D0*XSI(K) -
     1           81.D0*XSI(K)**2)
      C(1,4) = (1.D0 - ETA(J))*(1.D0 - ZTA(I))*(27.D0*XSI(K)**2 +
     1           9.D0*ETA(J)**2 + 18.D0*XSI(K) - 10.D0)
      C(1,5) = (1.D0 - ETA(J))*(1.D0 + ZTA(I))*(10.D0 + 18.D0*XSI(K) -
     1           27.D0*XSI(K)**2 - 9.D0*ETA(J)**2)
      C(1,6) = (1.D0 - ETA(J))*(1.D0 + ZTA(I))*(81.D0*XSI(K)**2
     1           -18.D0*XSI(K) - 27.D0)
      C(1,7) = (1.D0 - ETA(J))*(1.D0 + ZTA(I))*(27.D0 - 18.D0*XSI(K) -
     1           81.D0*XSI(K)**2)
      C(1,8) = (1.D0 - ETA(J))*(1.D0 + ZTA(I))*(27.D0*XSI(K)**2 +
     1           9.D0*ETA(J)**2 + 18.D0*XSI(K) - 10.D0)
      C(1,9) = (1.D0-3.D0*ETA(J))*(1.D0-ZTA(I))*(9.D0*ETA(J)**2-9.D0)
      C(1,10)= (1.D0+3.D0*ETA(J))*(1.D0-ZTA(I))*(9.D0*ETA(J)**2-9.D0)
      C(1,11)= -C(1,9)
      C(1,12)= -C(1,10)
```

```
      C(1,13)= (1.D0-3.D0*ETA(J))*(1.D0+ZTA(I))*(9.D0*ETA(J)**2-9.D0)
      C(1,14)= (1.D0+3.D0*ETA(J))*(1.D0+ZTA(I))*(9.D0*ETA(J)**2-9.D0)
      C(1,15)= -C(1,13)
      C(1,16)= -C(1,14)
      C(1,17)= (1.D0 + ETA(J))*(1.D0 - ZTA(I))*(10.D0 + 18.D0*XSI(K) -
     1         27.D0*XSI(K)**2 - 9.D0*ETA(J)**2)
      C(1,18)= (1.D0 + ETA(J))*(1.D0 - ZTA(I))*(81.D0*XSI(K)**2
     1         -18.D0*XSI(K) - 27.D0)
      C(1,19)= (1.D0 + ETA(J))*(1.D0 - ZTA(I))*(27.D0 - 18.D0*XSI(K) -
     1         81.D0*XSI(K)**2)
      C(1,20)= (1.D0 + ETA(J))*(1.D0 - ZTA(I))*(27.D0*XSI(K)**2 +
     1         9.D0*ETA(J)**2 + 18.D0*XSI(K) - 10.D0)
      C(1,21)= (1.D0 + ETA(J))*(1.D0 + ZTA(I))*(10.D0 + 18.D0*XSI(K) -
     1         27.D0*XSI(K)**2 - 9.D0*ETA(J)**2)
      C(1,22)= (1.D0 + ETA(J))*(1.D0 + ZTA(I))*(81.D0*XSI(K)**2
     1         -18.D0*XSI(K) - 27.D0)
      C(1,23)= (1.D0 + ETA(J))*(1.D0 + ZTA(I))*(27.D0 - 18.D0*XSI(K) -
     1         81.D0*XSI(K)**2)
      C(1,24)= (1.D0 + ETA(J))*(1.D0 + ZTA(I))*(27.D0*XSI(K)**2 +
     1         9.D0*ETA(J)**2 + 18.D0*XSI(K) - 10.D0)
      C(2,1) = (1.D0 - XSI(K))*(1.D0 - ZTA(I))*(10.D0 + 18.D0*ETA(J) -
     1         9.D0*XSI(K)**2 - 27.D0*ETA(J)**2)
      C(2,2) = (1.D0 - 3.D0*XSI(K))*(1.D0 -ZTA(I))*(9.D0*XSI(K)**2-9.D0)
      C(2,3) = (1.D0 + 3.D0*XSI(K))*(1.D0 -ZTA(I))*(9.D0*XSI(K)**2-9.D0)
      C(2,4) = (1.D0 + XSI(K))*(1.D0 - ZTA(I))*(10.D0 + 18.D0*ETA(J) -
     1         9.D0*XSI(K)**2 - 27.D0*ETA(J)**2)
      C(2,5) = (1.D0 - XSI(K))*(1.D0 + ZTA(I))*(10.D0 + 18.D0*ETA(J) -
     1         9.D0*XSI(K)**2 - 27.D0*ETA(J)**2)
      C(2,6) = (1.D0 - 3.D0*XSI(K))*(1.D0 +ZTA(I))*(9.D0*XSI(K)**2-9.D0)
      C(2,7) = (1.D0 + 3.D0*XSI(K))*(1.D0 +ZTA(I))*(9.D0*XSI(K)**2-9.D0)
      C(2,8) = (1.D0 + XSI(K))*(1.D0 + ZTA(I))*(10.D0 + 18.D0*ETA(J) -
     1         9.D0*XSI(K)**2 - 27.D0*ETA(J)**2)
      C(2,9) = (1.D0 - XSI(K))*(1.D0 -ZTA(I))*(81.D0*ETA(J)**2 -
     1         18.D0*ETA(J) - 27.D0)
      C(2,10)= (1.D0 - XSI(K))*(1.D0 - ZTA(I))*(27.D0 - 18.D0*ETA(J) -
     1         81.D0*ETA(J)**2)
      C(2,11)= (1.D0 + XSI(K))*(1.D0 -ZTA(I))*(81.D0*ETA(J)**2 -
     1         18.D0*ETA(J) - 27.D0)
      C(2,12)= (1.D0 + XSI(K))*(1.D0 - ZTA(I))*(27.D0 - 18.D0*ETA(J) -
     1         81.D0*ETA(J)**2)
      C(2,13)= (1.D0 - XSI(K))*(1.D0 +ZTA(I))*(81.D0*ETA(J)**2 -
     1         18.D0*ETA(J) - 27.D0)
      C(2,14)= (1.D0 - XSI(K))*(1.D0 + ZTA(I))*(27.D0 - 18.D0*ETA(J) -
     1         81.D0*ETA(J)**2)
      C(2,15)= (1.D0 + XSI(K))*(1.D0 +ZTA(I))*(81.D0*ETA(J)**2 -
     1         18.D0*ETA(J) - 27.D0)
      C(2,16)= (1.D0 + XSI(K))*(1.D0 + ZTA(I))*(27.D0 - 18.D0*ETA(J) -
     1         81.D0*ETA(J)**2)
      C(2,17)= (1.D0 - XSI(K))*(1.D0 - ZTA(I))*(27.D0*ETA(J)**2 +
     1         9.D0*XSI(K)**2 + 18.D0*ETA(J) - 10.D0)
      C(2,18)= -C(2,2)
```

```
      C(2,19)= -C(2,3)
      C(2,20)= (1.D0 + XSI(K))*(1.D0 - ZTA(I))*(27.D0*ETA(J)**2 +
     1         9.D0*XSI(K)**2 + 18.D0*ETA(J) - 10.D0)
      C(2,21)= (1.D0 - XSI(K))*(1.D0 + ZTA(I))*(27.D0*ETA(J)**2 +
     1         9.D0*XSI(K)**2 + 18.D0*ETA(J) - 10.D0)
      C(2,22)= (1.D0 - 3.D0*XSI(K))*(1.D0 +ZTA(I))*(9.D0-9.D0*XSI(K)**2)
      C(2,23)= (1.D0 + 3.D0*XSI(K))*(1.D0 +ZTA(I))*(9.D0-9.D0*XSI(K)**2)
      C(2,24)= (1.D0 + XSI(K))*(1.D0 + ZTA(I))*(27.D0*ETA(J)**2 +
     1         9.D0*XSI(K)**2 + 18.D0*ETA(J) - 10.D0)
      C(3,1) = (1.D0 - XSI(K))*(1.D0 - ETA(J))*(10.D0 - 9.D0*XSI(K)**2
     1         -9.D0*ETA(J)**2)
      C(3,2) = (1.D0 -3.D0*XSI(K))*(1.D0 - ETA(J))*(9.D0*XSI(K)**2-9.D0)
      C(3,3) = (1.D0 +3.D0*XSI(K))*(1.D0 - ETA(J))*(9.D0*XSI(K)**2-9.D0)
      C(3,4) = (1.D0 + XSI(K))*(1.D0 - ETA(J))*(10.D0 - 9.D0*XSI(K)**2
     1         -9.D0*ETA(J)**2)
      DO 26 N=1,4
   26 C(3,N+4) = -C(3,N)
      C(3,9) = (1.D0 -3.D0*ETA(J))*(1.D0 - XSI(K))*(9.D0*ETA(J)**2-9.D0)
      C(3,10)= (1.D0 +3.D0*ETA(J))*(1.D0 - XSI(K))*(9.D0*ETA(J)**2-9.D0)
      C(3,11)= (1.D0 -3.D0*ETA(J))*(1.D0 + XSI(K))*(9.D0*ETA(J)**2-9.D0)
      C(3,12)= (1.D0 +3.D0*ETA(J))*(1.D0 + XSI(K))*(9.D0*ETA(J)**2-9.D0)
      DO 27 N=9,12
   27 C(3,N+4) = -C(3,N)
      C(3,17)= (1.D0 - XSI(K))*(1.D0 + ETA(J))*(10.D0 - 9.D0*XSI(K)**2
     1         -9.D0*ETA(J)**2)
      C(3,18)= (1.D0 -3.D0*XSI(K))*(1.D0 + ETA(J))*(9.D0*XSI(K)**2-9.D0)
      C(3,19)= (1.D0 +3.D0*XSI(K))*(1.D0 + ETA(J))*(9.D0*XSI(K)**2-9.D0)
      C(3,20)= (1.D0 + XSI(K))*(1.D0 + ETA(J))*(10.D0 - 9.D0*XSI(K)**2
     1         -9.D0*ETA(J)**2)
      DO 28 N=17,20
   28 C(3,N+4) = -C(3,N)
C
C CALCULATE JACOBIAN MATRIX  J(3X3) = C(3X24) * XYZ(24X3)  AT 32 PT.
C
      DO 30 II=1,3
      DO 30 KK=1,3
      DJ(II,KK) = 0.D0
      DO 30 JJ=1,24
   30 DJ(II,KK) = DJ(II,KK) + C(II,JJ) * XYZ(JJ,KK) /64.D0
C
C FORM INVERSE J MATRIX  DJI(3X3)  FOR COORDINATE TRANSFORMATION
C
      DETJ    = DJ(1,1)*(DJ(2,2)*DJ(3,3) - DJ(2,3)*DJ(3,2))
     1         +DJ(1,2)*(DJ(2,3)*DJ(3,1) - DJ(2,1)*DJ(3,3))
     2         +DJ(1,3)*(DJ(3,2)*DJ(2,1) - DJ(2,2)*DJ(3,2))
      DJI(1,1) = (DJ(2,2)*DJ(3,3) - DJ(2,3)*DJ(3,2)) /DETJ
      DJI(1,2) = (DJ(3,2)*DJ(1,3) - DJ(3,3)*DJ(1,2)) /DETJ
      DJI(1,3) = (DJ(1,2)*DJ(2,3) - DJ(1,3)*DJ(2,2)) /DETJ
      DJI(2,1) = (DJ(2,3)*DJ(3,1) - DJ(2,1)*DJ(3,3)) /DETJ
      DJI(2,2) = (DJ(3,3)*DJ(1,1) - DJ(3,1)*DJ(1,3)) /DETJ
      DJI(2,3) = (DJ(1,3)*DJ(2,1) - DJ(1,1)*DJ(2,3)) /DETJ
```

```
      DJI(3,1) = (DJ(2,1)*DJ(3,2) - DJ(2,2)*DJ(3,1)) /DETJ
      DJI(3,2) = (DJ(3,1)*DJ(1,2) - DJ(3,2)*DJ(1,1)) /DETJ
      DJI(3,3) = (DJ(1,1)*DJ(2,2) - DJ(1,2)*DJ(2,1)) /DETJ
C FORM MATRIX  B(6X72), WHERE (B) = (BA)
      DO 40 N=1,6
      DO 40 L=1,72
   40 BA(N,L) = 0.D0
      DO 50 N=1,70,3
      L = (N - 1)/3 + 1
      BA(1,N) = (DJI(1,1)*C(1,L)+DJI(1,2)*C(2,L)+DJI(1,3)*C(3,L))/64.D0
      BA(4,N) = (DJI(2,1)*C(1,L)+DJI(2,2)*C(2,L)+DJI(2,3)*C(3,L))/64.D0
   50 BA(5,N) = (DJI(3,1)*C(1,L)+DJI(3,2)*C(2,L)+DJI(3,3)*C(3,L))/64.D0
      DO 60 N=2,71,3
      L = (N - 2)/3 + 1
      BA(2,N) = (DJI(2,1)*C(1,L)+DJI(2,2)*C(2,L)+DJI(2,3)*C(3,L))/64.D0
      BA(4,N) = (DJI(1,1)*C(1,L)+DJI(1,2)*C(2,L)+DJI(1,3)*C(3,L))/64.D0
   60 BA(6,N) = (DJI(3,1)*C(1,L)+DJI(3,2)*C(2,L)+DJI(3,3)*C(3,L))/64.D0
      DO 70 N=3,72,3
      L = (N - 3)/3 + 1
      BA(3,N) = (DJI(3,1)*C(1,L)+DJI(3,2)*C(2,L)+DJI(3,3)*C(3,L))/64.D0
      BA(5,N) = (DJI(1,1)*C(1,L)+DJI(1,2)*C(2,L)+DJI(1,3)*C(3,L))/64.D0
   70 BA(6,N) = (DJI(2,1)*C(1,L)+DJI(2,2)*C(2,L)+DJI(2,3)*C(3,L))/64.D0
      RETURN
      END
```

```
      SUBROUTINE  SHPFNT
C
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    *                                                                 *
C    *  SUBROUTINE SHPFNT CALCULATES THE SHAPE FUNCTION VECTOR         *
C    *                                                                 *
C    *  THIS SUBROUTINE IS CALLED BY -                                 *
C    *        ELSTIF                                                   *
C    *                                                                 *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON  / STIFIN /  D(6,6,24), XNVCT(24), ETM(9), BA(6,72),
     1    C(3,24), XYZ(24,3), ALFTMP(6), XSIK, ETAJ, ZTAI
      XP = 1.D0 + XSIK
      XM = 1.D0 - XSIK
      YP = 1.D0 + ETAJ
      YM = 1.D0 - ETAJ
      ZP = 1.D0 + ZTAI
      ZM = 1.D0 - ZTAI
      XP3 = 1.D0 + 3.D0*XSIK
      XM3 = 1.D0 - 3.D0*XSIK
      YP3 = 1.D0 + 3.D0*ETAJ
      YM3 = 1.D0 - 3.D0*ETAJ
      XNVCT(1) = XM*YM*ZM*(9.D0*(XSIK**2+ETAJ**2)  -  10.D0) / 64.D0
      XNVCT(2) = 9.D0*XM3*YM*ZM* (1.D0-XSIK**2) / 64.D0
      XNVCT(3) = XNVCT(2) * XP3 / XM3
      XNVCT(4) = XNVCT(1) * XP  / XM
      XNVCT(5) = XNVCT(1) * ZP  / ZM
      XNVCT(6) = XNVCT(2) * ZP  / ZM
      XNVCT(7) = XNVCT(3) * ZP  / ZM
      XNVCT(8) = XNVCT(4) * ZP  / ZM
      XNVCT(9) = 9.D0 * XM*YM3*ZM*( 1.D0-ETAJ**2) / 64.D0
      XNVCT(10) = XNVCT( 9) * YP3 / YM3
      XNVCT(11) = XNVCT( 9) * XP  / XM
      XNVCT(12) = XNVCT(11) * YP3 / YM3
      XNVCT(13) = XNVCT( 9) * ZP  / ZM
      XNVCT(14) = XNVCT(13) * YP3 / YM3
      XNVCT(15) = XNVCT(13) * XP  / XM
      XNVCT(16) = XNVCT(14) * XP  / XM
      XNVCT(17) = XNVCT( 1) * YP  / YM
      XNVCT(18) = XNVCT( 2) * YP  / YM
      XNVCT(19) = XNVCT(18) * XP3 / XM3
      XNVCT(20) = XNVCT(17) * XP  / XM
      XNVCT(21) = XNVCT(17) * ZP  / ZM
      XNVCT(22) = XNVCT(18) * ZP  / ZM
      XNVCT(23) = XNVCT(22) * XP3 / XM3
      XNVCT(24) = XNVCT(21) * XP  / XM
      RETURN
      END
```

```
      SUBROUTINE  DMAT
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                               *
C   *  SUBROUTINE DMAT CALCULATES THE ELASTIC MATRIX AND PERFORMS A *
C   *  ROTATIONAL TRANSFORMATION ON THE ELASTIC MATRIX              *
C   *                                                               *
C   *  THIS SUBROUTINE IS CALLED BY -                               *
C   *        ELAS                                                   *
C   *                                                               *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2  IX, ICODE, GNMAT, MTLND, LDBC
      COMMON / GENL /  XINIT, EPS, AMBTMP,
     1       ICLASS, NEL, NGNP, NGLDF, NMTL, NTYEL, LIMIT, NM, NDBC
      COMMON  /NODELM/  X(1015), Y(1015), Z(1015), UX(1015), UY(1015),
     1  UZ(1015), TMPND(1015), FBC(3045), DBC(3045), TBC( 1,72),
     2   ICODE(1015), IX(144,27), GNMAT(144,72),MTLND(1015),LDBC(1015)
      COMMON  /MATL /    E(9,9,10),           FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9), TMPEL(9,10), NTMP(9)
      COMMON  / STIFIN /  D(6,6,24), XNVCT(24), ETM(9), BA(6,72),
     1    C(3,24), XYZ(24,3), ALFTMP(6), XSIK, ETAJ, ZTAI
      COMMON / INDX /   INEL, IGNP, ILNP,IMTL
      DIMENSION       T(6,6),TD(6,6), TT(6,6), TMPCOF(6), DTMP(6,6)
      DO 10 I=1,6
      DO 10 J=1,6
      T(I,J) = 0.D0
      D(I,J,ILNP) = 0D0
   10 TD(I,J) = 0.D0
      XNU21 = ETM(4) * ETM(2) / ETM(1)
      XNU31 = ETM(5) * ETM(3) / ETM(1)
      XNU32 = ETM(6) * ETM(3) / ETM(2)
      FACT=1.D0-ETM(4)*(XNU21+ETM(6)*XNU31)-ETM(5)*(XNU31+XNU32*XNU21)-
     1     ETM(6)*XNU32
      D(1,1,ILNP)= ETM(1) * (1.D0 - ETM(6) * XNU32) / FACT
      D(1,2,ILNP) = ETM(2) * (ETM(4) + ETM(5) * XNU32) / FACT
      D(1,3,ILNP) = ETM(3) * (ETM(5) + ETM(4) * ETM(6)) / FACT
      D(2,1,ILNP) = D(1,2,ILNP)
      D(2,2,ILNP) = ETM(2) * (1.D0 - ETM(5) * XNU31) / FACT
      D(2,3,ILNP) = ETM(3) *(ETM(6) + ETM(5)* XNU21) / FACT
      D(3,1,ILNP) = D(1,3,ILNP)
      D(3,2,ILNP) = D(2,3,ILNP)
      D(3,3,ILNP) = ETM(3)* (1.D0 - ETM(4) * XNU21) / FACT
      D(4,4,ILNP) = ETM(7)
      D(5,5,ILNP) = ETM(8)
      D(6,6,ILNP) = ETM(9)
      ALFTMP(1) = ALFA1(IMTL)
      ALFTMP(2) = ALFA2(IMTL)
      ALFTMP(3) = ALFA3(IMTL)
      DO 60 I=4,6
```

```
60 ALFTMP(I) = 0D0
   IF( DABS(FIBORT(IMTL)) .LT.  .5D-14 )  GO TO 50
   FIBOR = FIBORT(IMTL) * 3.14159265358979320D0 / 180.D0
   T(1,1) = DCOS(FIBOR )**2
   T(1,2) = DSIN(FIBOR )**2
   T(4,1) = DCOS(FIBOR )  *  DSIN(FIBOR )
   T(1,4) = -2.D0 *T(4,1)
   T(2,1) = T(1,2)
   T(2,2) = T(1,1)
   T(2,4) = -T(1,4)
   T(3,3) = 1.D0
   T(4,2) = -T(4,1)
   T(4,4) = T(1,1) - T(1,2)
   T(5,5) = DCOS(FIBOR )
   T(6,5) = DSIN(FIBOR )
   T(5,6) = -T(6,5)
   T(6,6) = T(5,5)
   DO 70 I=1,6
   TMPCOF(I) = 0D0
   DO 70 J=1,6
70 TMPCOF(I) = TMPCOF(I) + T(I,J) * ALFTMP(J)
   DO 90 I=1,6
90 ALFTMP(I) = TMPCOF(I)
   DO 20 I=1,6
   DO 20 J=1,6
   DO 20 K=1,6
20 TD(I,J) = TD(I,J) + T(I,K)*D(K,J,ILNP)
   DO 80 I=1,6
   DO 80 J=1,6
80 TT(J,I) = T(I,J)
   DO 30 I=1,6
   DO 30 J=1,6
   DTMP(I,J) = 0D0
   DO 30 K=1,6
30 DTMP(I,J) = DTMP(I,J) + TD(I,K) * TT(K,J)
   DO 40 I=1,6
   DO 40 J=1,6
40 D(I,J,ILNP) = DTMP(I,J)
50 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE    FBCDBC
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE FBCDBC FORMS A FORCE AND A DISPLACEMENT ARRAY         *
C   *                                                                   *
C   *  THIS SUBROUTINE IS CALLED BY -                                   *
C   *        MAIN                                                       *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2  IX, ICODE, GNMAT, MTLND, LDBC
      COMMON / GENL /  XINIT, EPS, AMBTMP,
     1        ICLASS, NEL, NGNP, NGLDF, NMTL, NTYEL, LIMIT, NM, NDBC
      COMMON  /NODELM/  X(1015), Y(1015), Z(1015), UX(1015), UY(1015),
     1  UZ(1015), TMPND(1015), FBC(3045), DBC(3045), TBC( 1,72),
     2   ICODE(1015), IX(144,27), GNMAT(144,72),MTLND(1015),LDBC(1015)
      COMMON  /MATL /   E(9,9,10),          FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9), TMPEL(9,10), NTMP(9)
      IDBC = 0
      DO 7  I=1,NGNP
      IF(ICODE(I) .EQ. 1 .OR. ICODE(I) .EQ. 3)  GO TO 2
      IF(ICODE(I) .EQ. 2)  GO TO 3
      IF(ICODE(I) .EQ. 7)  GO TO 4
      FBC(3*I-2) = UX(I)
      IF(ICODE(I) .EQ. 4)  GO TO 3
      IF(ICODE(I) .EQ. 6)  GO TO 5
    2 FBC(3*I-1) = UY(I)
      IF(ICODE(I) .EQ. 3)  GO TO 4
      IF(ICODE(I) .EQ. 5)  GO TO 6
    3 FBC(3*I  ) = UZ(I)
      IF(ICODE(I) .EQ. 0)  GO TO 7
      IF(ICODE(I) .EQ. 4)  GO TO 5
    4 IDBC = IDBC + 1
      LDBC(IDBC) = 3*I-2
      DBC(3*I-2) = UX (I)
      IF(ICODE(I) .EQ. 1)  GO TO 7
      IF(ICODE(I) .EQ. 3)  GO TO 6
    5 IDBC = IDBC + 1
      LDBC(IDBC) = 3*I-1
      DBC(3*I-1) = UY (I)
      IF(ICODE(I) .EQ. 2 .OR. ICODE(I) .EQ. 4)  GO TO 7
    6 IDBC = IDBC + 1
      LDBC(IDBC) = 3*I
      DBC(3*I  ) = UZ (I)
    7 CONTINUE
      NDBC = IDBC
      RETURN
      END
```

```
C   MAIN PROGRAM STEP 3 (ITERATION IN CORE VERSION)                      ESC
C                                                                        ESC
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESC
C   *                                                                  * ESC
C   *  STEP 3 PERFORMS FIVE FUNCTIONS                                  * ESC
C   *                                                                  * ESC
C   *  1.    COMBINES NON-ZERO DISPLACEMENT BOUNDARY CONDITIONS WITH   * ESC
C   *        THE FORCE VECTOR .                                        * ESC
C   *                                                                  * ESC
C   *  2.    SOLVES THE SYSTEM OF LINEAR EQUATIONS BY MINIMIZING THE   * ESC
C   *        TOTAL POTENTIAL ENERGY                                    * ESC
C   *                                                                  * ESC
C   *  3.    CALCULATES AND PRINTS THE STRAIN ENERGY AND OTHER         * ESC
C   *        CONVERGENCE PARAMETERS                                    * ESC
C   *                                                                  * ESC
C   *  4.    PRINTS THE COORDINATE AND THREE (X,Y,&Z) DISPLACEMENTS    * ESC
C   *        AT EACH NODE                                              * ESC
C   *                                                                  * ESC
C   *  5.    WRITES DISPLACEMENTS ON DISK AND/OR CARDS                 * ESC
C   *                                                                  * ESC
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESC
C                                                                        ESC
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESC
C   *                                                                  * ESC
C   *  VARIABLE DEFINITIONS AND DIMENSIONS FOR STEP 3                  * ESC
C   *                                                                  * ESC
C   *  ACB - - - - - - - CHANGE IN STRAIN ENERGY OVER STRAIN ENERGY    * ESC
C   *                                                                  * ESC
C   *  B(NGLDF) - - - - - FORCE VECTOR                                 * ESC
C   *                                                                  * ESC
C   *  BNRM - - - - - - - MAGNITUDE OF THE FORCE VECTOR                * ESC
C   *                                                                  * ESC
C   *  DBC(MM) - - - - -  NON-ZERO DISPLACEMENT BOUNDARY CONDITIONS    * ESC
C   *                                                                  * ESC
C   *  DELE - - - - - - - CHANGE IN STRAIN ENERGY                      * ESC
C   *                                                                  * ESC
C   *  DELXNR - - - - - - MAGNITUDE OF THE CHANGE IN DISPLACEMENT      * ESC
C   *                     VECTOR                                       * ESC
C   *                                                                  * ESC
C   *  ENGY1 - - - - - -  STRAIN ENERGY                                * ESC
C   *                                                                  * ESC
C   *  ENGY2 - - - - - -  STRAIN ENERGY                                * ESC
C   *                                                                  * ESC
C   *  EPS - - - - - - -  ENERGY CONVERGENCE TEST PARAMETER            * ESC
C   *                                                                  * ESC
C   *  G(NGLDF) - - - - - RESULTING VECTOR FROM KX OR KP MATRIX-       * ESC
C   *                     VECTOR PRODUCT                               * ESC
C   *                                                                  * ESC
C   *  GNMAT(NEL,72) - -  RELATES LOCAL AND GLOBAL DEGREES-OF-FREEDOM  * ESC
C   *                                                                  * ESC
C   *  IX26(NEL) - - - -  ELEMENT TYPE FOR EACH ELEMENT                * ESC
```

```
C   *                                                            * ESC  520
C   *  KEMAT(NTYEL,2628)- MATRIX OF THE UPPER SYMMETRIC PORTION OF THE* ESC  530
C   *                     UNIQUE ELEMENT STIFFNESS MATRICES - EACH   * ESC  540
C   *                     ELEMENT STIFFNESS MATRIX IS STORED AS A ROW * ESC  550
C   *                     IN KEMAT                                   * ESC  560
C   *                                                                * ESC  570
C   *  KOUNT - - - - - -  TOTAL NUMBER OF ITERATIONS                 * ESC  580
C   *                                                                * ESC  590
C   *  LDBC(NDBC) - - - - INDEX FOR DISPLACEMENT BOUNDARY CONDITIONS * ESC  600
C   *                                                                * ESC  610
C   *  LLDBC(MM) - - - -  INDEX NUMBERS FOR NON-ZERO DISPLACEMENT    * ESC  620
C   *                     BOUNDARY CONDITIONS                        * ESC  630
C   *                                                                * ESC  640
C   *  MM - - - - - - - - NUMBER OF NON-ZERO DISPLACEMENT BOUNDARY   * ESC  650
C   *                     CONDITIONS                                 * ESC  660
C   *                                                                * ESC  670
C   *  NDBC - - - - - - - NUMBER OF DISPLACEMENT BOUNDARY CONDITIONS * ESC  680
C   *                                                                * ESC  690
C   *  NEL - - - - - - -  NUMBER OF ELEMENTS                         * ESC  700
C   *                                                                * ESC  710
C   *  NGLDF - - - - - -  NUMBER OF DEGREES-OF-FREEDOM (GLOBAL SYSTEM)* ESC  720
C   *                                                                * ESC  730
C   *  NTYEL - - - - - -  NUMBER OF UNIQUE ELEMENTS                  * ESC  740
C   *                                                                * ESC  750
C   *  P(NGLDF) - - - - - CORRECTION VECTOR IN CONJUGATE GRADIENT    * ESC  760
C   *                     ROUTINE                                    * ESC  770
C   *                                                                * ESC  780
C   *  R(NGLDF) - - - - - RESIDUE VECTOR IN CONJUGATE GRADIENT       * ESC  790
C   *                     ROUTINE                                    * ESC  800
C   *                                                                * ESC  810
C   *  RNRM - - - - - - - MAGNITUDE OF THE RESIDUE VECTOR            * ESC  820
C   *                                                                * ESC  830
C   *  UX(NGNP) - - - - - DISPLACEMENTS IN THE X-DIRECTION           * ESC  840
C   *                                                                * ESC  850
C   *  UY(NGNP) - - - - - DISPLACEMENTS IN THE Y-DIRECTION           * ESC  860
C   *                                                                * ESC  870
C   *  UZ(NGNP) - - - - - DISPLACEMENTS IN THE Z-DIRECTION           * ESC  880
C   *                                                                * ESC  890
C   *  X(NGLDF) - - - - - DISPLACEMENT VECTOR                        * ESC  900
C   *                                                                * ESC  910
C   *  XCORD(NGNP) - - -  X-COORDINATE (GLOBAL SYSTEM)               * ESC  920
C   *                                                                * ESC  930
C   *  YCORD(NGNP) - - -  Y-COORDINATE (GLOBAL SYSTEM)               * ESC  940
C   *                                                                * ESC  950
C   *  ZCORD(NGNP) - - -  Z-COORDINATE (GLOBAL SYSTEM)               * ESC  960
C   *                                                                * ESC  970
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESC  980
C                                                                      ESC  990
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESC 1000
C   *                                                                * ESC 1010
C   *  LOGICAL SWITCH INFORMATION ( ALL SWITCHES INITIALLY .FALSE.)  * ESC 1020
```

```
C
C    *  SW(1)  .TRUE.     CONVERGED TO A SOLUTION                          *
C    *         .FALSE.    DID NOT CONVERGE TO A SOLUTION                   *
C    *         SET IN     CGJRD                                            *
C    *         TESTED IN  MAIN                                             *
C    *                                                                     *
C    *  SW(2)  .TRUE.     CHANGE IN STRAIN ENERGY/STRAIN ENERGY .LT. EPS*
C    *         .FALSE.    CHANGE IN STRAIN ENERGY/STRAIN ENERGY .GE. EPS*
C    *         SET IN     CGJRD                                            *
C    *         TESTED IN CGJRD                                             *
C    *                                                                     *
C    *  SW(3)  .TRUE.     EXCEEDED MAXIMUM NUMBER OF ITERATIONS            *
C    *         .FALSE.    DID NOT EXCEED MAXIMUM NUMBER OF ITERATIONS      *
C    *         SET IN     CGJRD                                            *
C    *         TESTED IN CGJRD                                             *
C    *                                                                     *
C    *  SW(4)  .TRUE.     MAGNITUDE OF THE RESIDUE VECTOR .LT. 1.00        *
C    *         .FALSE.    MAGNITUDE OF THE RESIDUE VECTOR .GE. 1.00        *
C    *         SET IN     CGJRD                                            *
C    *         TESTED IN CGJRD                                             *
C    *                                                                     *
C    *  SW(8)  .TRUE.     INITIAL DISPLACEMENTS READ FROM DISK             *
C    *         .FALSE.    INITIAL DISPLACEMENTS READ FROM CARDS            *
C    *         SET IN     MAIN                                             *
C    *         TESTED IN CGJRD                                             *
C    *  SW(5), SW(6), SW(7), SW(9), SW(10), SW(11) & SW(12) NOT USED       *
C    *                                                                     *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT  REAL*8  (A-H,O-Z)
      REAL*8KEMAT
      LOGICAL*1   SW
      INTEGER*2   GNMAT,LDBC,LLDBC, IX26
      COMMON / KEGN /   KEMAT(3,2628), GNMAT( 64,72),IX26( 65)
      COMMON / CGVECT / G(1575),X(1575),B(1575),P(1575),R(1575)
      COMMON /BC/  DBC(200),  LLDBC(200),  LDBC(700),  NDBC
      COMMON /SCALAR/   EPS, ACB, XNRM, BNRM, RNRM,DELXNR,
     1           DELE, ENGY1, ENGY2,
     2        NGLDF, LIMIT, NEL, KOUNT, ITYEL, ITRLMT, MM
      COMMON / HEAD /  HED(10),ICRD,LIST,IPAGE,LINE
      COMMON /SWITCH /       SW(12)
C
C   CAUTION:  DO NOT DIMENSION UX GREATER THAN 1314 - SEE EQUIVALENCE
C             STATEMENT BELOW
C
      DIMENSION  XCORD(525), YCORD(525), ZCORD(525),
     1        UX(525), UY(525), UZ(525)
      EQUIVALENCE  (XCORD(1),G(1)),  (YCORD(1) ,B(1)),  (ZCORD(1), P(1))
      EQUIVALENCE (KEMAT(1,1),UX(1)),(KEMAT(1, 1315),UY(1)),
     1             (R(1), UZ(1))
  100 FORMAT(2I5,2F10.0, I5 )
```

```
  102 FORMAT(5216)
  201 FORMAT('OMAXIMUM NUMBER OF ITERATIONS FOR THIS RUN IS' , I5)
  301 FORMAT('ODELTA STRAIN ENERGY / STRAIN ENERGY DID CONVERGE TO '
     1     , E14.7,' AFTER' , I6, ' ITERATIONS AND ACCURATE TO',E14.7)
  302 FORMAT(' TIME IN CG SUBROUTINE IS ' , F7.2, ' SECONDS')
  303 FORMAT('ODELTA STRAIN ENERGY / STRAIN ENERGY DID NOT CONVERGE TO '
     1     , E14.7,' AFTER' , I6, ' ITERATIONS BUT ACCURATE TO',E14.7)
  304 FORMAT('O', G14.7, ' IS THE INITIAL GUESS FOR ALL DISPLACEMENTS')
  305 FORMAT ('OTHE INITIAL GUESSES FOR DISPLACEMENTS ARE READ FROM CARD
     1S AND MULTIPLIED BY A FACTOR OF' , G14.7)
  306 FORMAT ('OTHE INITIAL GUESSES FOR DISPLACEMENTS ARE READ FROM CARD
     1S AND THE Z-DISPLACEMENTS ARE MULTIPLIED BY A FACTOR OF', G14.7)
  307 FORMAT('OTHE INITIAL GUESSES FOR DISPLACEMENTS ARE READ FROM DISK'
     1    )
  308 FORMAT('OSTRAIN ENERGY',  T50, E14.7 ,' K IN-LBS' /
     1        ' CHANGE IN STRAIN ENERGY',  T50,  E14.7 ,' K IN-LBS' /
     2        ' MAGNITUDE OF THE RESIDUE VECTOR',  T50, E14.7 ,' K LBS'/
     3        ' MAGNITUDE OF THE FORCE VECTOR',  T50, E14.7 ,' K LBS'/
     4        ' MAGNITUDE OF THE CHANGE IN DISPLACEMENT VECTOR',T50,E14.7
     5,' IN'/' MAGNITUDE OF THE DISPLACEMENT VECTOR',  T50, E14.7,' IN')
 2000 FORMAT('O',7X,'NODE',13X,'X-COORD',13X, 'Y-COORD',13X, 'Z-COORD',
     1   .   9X, 'X-DISPL', 13X, 'Y-DISPL',13X, 'Z-DISPL' /
     2      27X, 'INS', 17X, 'INS',17X, 'INS', 13X, 'INS', 17X, 'INS',
     3         17X, 'INS' / )
 2001 FORMAT(6X, I5, 3F20.5, 3E20.7)
      DEFINE FILE 3(55,6500,U,IDXDA)
      IUDA = 3
      LIST = 6
C
C   READ CARD DATA FOR STEP 3
C
      READ(5,100)  INTXMD, ITRLMT, FACTOR, EPS, IDSPL
      WRITE(LIST,201)  ITRLMT
C
C   READ DATA GENERATED IN STEP 2
C
      READ (IUDA'1)      NEL, NGLDF,       NDBC, NTYEL, LIMIT, NGNP,NMTL,
     1     HED, IPAGE, AMBTMP
      READ (IUDA'4)     (  B(J),J=1,NGLDF)
     1   , SW, NUCNV,   (LDBC(J),J=1,NDBC)
      READ (IUDA'5)     (  P(J),J=1,NGLDF)
     1                                     , (IX26(J),J=1,NEL)
      READ (IUDA'6)    ((GNMAT(I,J),J=1,72),I=1,NEL)
C
C   READ UNIQUE ELEMENT STIFFNESS MATRICES
C
      DO 44 I=1,NTYEL
   44 READ(IUDA'IDXDA)  (KEMAT(I,J),J=1,2628)
      IDXDA = NTYEL+7
C
C   DETERMINE INITIAL GUESS FOR THE DISPLACEMENT VECTOR
```

```
C
      GO TO ( 21,22,23,24), INTXMD
   21 DO 50 I=1,NGLDF
   50 X(I) = FACTOR
      WRITE(LIST,304)  FACTOR
      GO TO 29
   22 READ(5,102) (X(J),J=1,NGLDF)
      DO 61 I=1,NGLDF
   61 X(I) = FACTOR*X(I)
      WRITE(LIST,305)  FACTOR
      GO TO 29
   23 READ(5,102)  (X(J),J=1,NGLDF)
      DO 10 I=1,NGNP
   10 X(3*I) = FACTOR*X(3*I)
      WRITE(LIST,306)  FACTOR
      GO TO 29
   24 READ (IUDA'IDXDA)SW,MM, (DBC(J),J=1,MM), (LLDBC(J),J=1,MM), KOUNT,
     1            (X(J),J=1,NGLDF)
      SW(8) = .TRUE.
      SW(3) = .FALSE.
   35 IF( .NOT. SW(10) )  GO TO 33
      MM = 0
   33 READ (IUDA'IDXDA)    (B(J),J=1,NGLDF)
      WRITE(LIST,307)
      GO TO 34
   29 CONTINUE
C
C   CONVERT INITIAL DISPLACEMENT BOUNDARY CONDITIONS TO FORCE BOUNDARY
C   CONDITIONS
C
      MM = 0
      DO 60 I=1,NGLDF
      IF( DABS(P(I)) .LT. 1.D-14)  GO TO 60
      MM = MM + 1
      DBC(MM) = P(I)
      LLDBC(MM) = I
   60 CONTINUE
      DO 13 I=1,NGLDF
   13 G(I) = 0.D0
      DO 15 INEL=1,NEL
      ITEL = IX26(INEL)
   15 CALL GVT(GNMAT, KEMAT, G, P, INEL, ITEL)
      DO 80 I=1,NGLDF
   80 B(I) = B(I) - G(I)
   34 CALL TIMON
C
C   MINIMIZE TOTAL POTENTIAL ENERGY
C
      CALL CGJRD
      CALL TIMECK(ITIME)
      SEC = ITIME/100.
```

```
C
C   WRITE FORCE AND DISPLACEMENT DATA ON DISK FOR USE IN FUTURE RUNS
C
      IF( MM .NE. 0 )  GO TO 31
      SW(10) = .TRUE.
      MM = 1
      DBC(1) = 0D0
   31 IDXDA = NTYEL + 7
      WRITE(IUDA'IDXDA)SW,MM, (DBC(J),J=1,MM), (LLDBC(J),J=1,MM), KOUNT,
     1          (X(J),J=1,NGLDF)
      WRITE(IUDA'IDXDA)    (B(J),J=1,NGLDF)
C
C   PRINT STRAIN ENERGY AND CONVERGENCE DATA
C
      IF(SW(1)) GO TO 20
      WRITE(LIST,303)  EPS, KOUNT, ACB
      GO TO 9
   20 WRITE(LIST,301)  EPS, KOUNT, ACB
    9 WRITE(LIST,302)  SEC
      WRITE(LIST,308)  ENGY1, DELE, RNRM, BNRM, DELXNR, XNRM
      IF(IDSPL .EQ. 0)  GO TO 32
C
C   PRINT AND/OR PUNCH THE DISPLACEMENT VECTOR
C
C
      IF(IDSPL .EQ. 2)  GO TO 43
      READ(IUDA'3)(XCORD(J),J=1,NGNP), (YCORD(J),J=1,NGNP),
     1             (ZCORD(J),J=1,NGNP)
      CALL TITLE
      WRITE(LIST,2000)
      DO 30 I=1,NGNP
      IF(LINE .LT. 48) GO TO 94
      CALL TITLE
      WRITE(LIST,2000)
   94 CONTINUE
      LINE = LINE + 1
      UX(I) = X(3*I-2)
      UY(I) = X(3*I-1)
      UZ(I) = X(3*I)
   30 WRITE(LIST,2001)  I,  XCORD(I) , YCORD(I), ZCORD(I),
     1                UX(I), UY(I), UZ(I)
      IF(IDSPL .EQ. 1)  GO TO 32
   43 WRITE(7,102)  (X(I),I=1,NGLDF)
   32 STOP
      END
```

```
      SUBROUTINE  CGJRD
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   *  SUBROUTINE CGJRD MINIMIZES THE TOTAL POTENTIAL ENERGY BY THE   *
C   *  CONJUGATE GRADIENT METHOD AND TESTS THE STRAIN ENERGY FOR      *
C   *  CONVERGENCE                                                    *
C   *                                                                 *
C   *  THIS SUBROUTINE IS CALLED BY -                                 *
C   *        MAIN                                                     *
C   *                                                                 *
C   *  THIS SUBROUTINE CALLS                                          *
C   *        GVT                                                      *
C   *                                                                 *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      IMPLICIT  REAL*8  (A-H,O-Z)
      REAL*8KEMAT
      INTEGER*2   GNMAT,LDBC,LLDBC, IX26
      LOGICAL*1    SW
      COMMON / KEGN /    KEMAT(3,2628), GNMAT( 64,72),IX26( 65)
      COMMON / CGVECT / G(1575),X(1575),B(1575),P(1575),R(1575)
      COMMON /SCALAR/    EPS, ACB, XNRM, BNRM, RNRM,DELXNR,
     1           DELE, ENGY1, ENGY2,
     2        NGLDF, LIMIT, NEL, KOUNT, ITYEL, ITRLMT, MM
      COMMON /BC/  DBC(200),  LLDBC(200),  LDBC(700),  NDBC
      COMMON /SWITCH /       SW(12)
      ITKNT = 0
      IF(SW(8) )  GO TO 2
      KOUNT = 0
      BETA = 0.D0
C
C   RESTART ITERATIVE PROCESS BY FINDING NEW RESIDUE VECTOR
C
    2 DO 10 I = 1,NGLDF
   10 G(I) = 0.D0
      DO 70 I=1,NDBC
      B(LDBC(I)) = 0.D0
   70 X(LDBC(I)) = 0.D0
      DO 31 INEL=1,NEL
      ITEL = IX26(INEL)
   31 CALL GVT(GNMAT, KEMAT, G, X, INEL, ITEL)
      DO 32 I=1,NDBC
   32 G(LDBC(I)) = 0.D0
      RNRM1 = 0.D0
      DO 20 I = 1,NGLDF
      P(I) = B(I) - G(I)
      R(I) = P(I)
   20 RNRM1 = RNRM1 + R(I) * R(I)
C
```

```
C   CALCULATE STRAIN ENERGY
C
      IF( .NOT. (SW(3) .OR. SW(4))) GO TO 5
      IF(MM .EQ. 0) GO TO 95
      DO 90 I=1,MM
   90 X(LLDBC(I)) = DBC(I)
   95 ENGY1 = 0.D0
      DO 17 I=1,NGLDF
   17 G(I) = 0.D0
      DO 34 INEL=1,NEL
      ITEL = IX26(INEL)
   34 CALL GVT(GNMAT, KEMAT, G, X, INEL, ITEL)
      DO 92 I=1,NGLDF
   92 ENGY1 = ENGY1 + X(I)   * G(I)
      ENGY1 = .5D0 * ENGY1
      ENGY2 = ENGY1
C
C   FIND ALFA
C
    5 DO 35 I=1,NGLDF
   35 G(I) = 0.D0
      DO 33 INEL=1,NEL
      ITEL = IX26(INEL)
   33 CALL GVT(GNMAT, KEMAT, G, P, INEL, ITEL)
      DO 36 I=1,NDBC
   36 G(LDBC(I)) = 0.D0
      PKP = 0.D0
      DO 30 I = 1,NGLDF
   30 PKP = PKP + P(I) * G(I)
      ALFA = RNRM1 / PKP
      IF(SW(3)) GO TO 25
C
C   CORRECT THE DISPLACEMENT VECTOR
C
      DO 40 I = 1,NGLDF
   40 X(I) = X(I) + ALFA * P(I)
      ITKNT = ITKNT + 1
      KOUNT = KOUNT + 1
      IF(ITKNT .LT. ITRLMT) GO TO 7
      SW(3) = .TRUE.
      GO TO 2
    7 IF(SW(4)) GO TO 12
      IF (RNRM1 .GT. 1.) GO TO 11
      SW(4) = .TRUE.
      GO TO 2
   11 RNRM2 = 0.D0
C
C   FIND NEW RESIDUE VECTOR AND NEW P VECTOR
C
      DO 50 I = 1,NGLDF
      R(I) = R(I) - ALFA*G(I)
```

```
   50 RNRM2 = RNRM2 + R(I) * R(I)
      BETA = RNRM2/RNRM1
      DO 60  I = 1,NGLDF
   60 P(I) =  R(I) + BETA * P(I)
      RNRM1 = RNRM2
      GO TO 5
C
C    FIND CHANGE IN STRAIN ENERGY FROM ALFA & RNRM1
C
   12 DELE = .5D0 * ALFA * RNRM1
      ENGY2 = ENGY2 - DELE
      IF(DELE/ENGY2 .LT. EPS)  GO TO 14
      SW(2) = .FALSE.
      GO TO 11
   14 IF(SW(2)) GO TO 15
      SW(2) = .TRUE.
      GO TO 2
   15 SW(1) = .TRUE.
C
C    CALCULATE CONVERGENCE PARAMETERS
C
   25 XNRM = 0.D0
      BNRM = 0.D0
      DELXNR = 0.D0
      DO 16 I=1,NGLDF
      XNRM = XNRM + X(I) * X(I)
      BNRM = BNRM + B(I) * B(I)
   16 DELXNR = DELXNR + P(I) * P(I)
      XNRM = DSQRT(XNRM)
      BNRM = DSQRT(BNRM)
      RNRM = DSQRT(RNRM1)
      DELXNR = ALFA * DSQRT(DELXNR)
      IF( .NOT. SW(2))  DELE = .5D0 * ALFA * RNRM1
      ACB = DELE / ENGY2
      RETURN
      END
```

```
      SUBROUTINE TITLE
C
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    *                                                                   *
C    *  SUBROUTINE TITLE PRINTS THE HEADING ON EACH PAGE                 *
C    *  THIS SUBROUTINE IS CALLED BY -                                   *
C    *       MAIN                                                        *
C    *                                                                   *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / HEAD /  HED(10),ICRD,LIST,IPAGE,LINE
  100 FORMAT (1H1,'FEM 72-DOF GENERAL HEXAHEDRONS THERMO-ELASTIC, VARYIN
     1G MATERIAL PROPERTIES, DANA', 9X, 'PAGE', I3)
  101 FORMAT (1H0,10A8 )
      LIST = 6
      IWRT = 6
      WRITE (LIST,100) IPAGE
      WRITE (LIST,101) HED
      IPAGE= IPAGE +1
      LINE = 0
      RETURN
      END
```

```
      SUBROUTINE GVT(GNMAT,KEMAT,G,P,INEL, ITEL)
C
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    *                                                                   *
C    *  SUBROUTINE GVT FORMS THE MATRIX-VECTOR PRODUCT KP=G WHERE K      *
C    *  REPRESENTS THE NON-FORMED GLOBAL STIFFNESS MATRIX                *
C    *                                                                   *
C    *  THIS SUBROUTINE IS CALLED BY -                                   *
C    *        MAIN                                                       *
C    *        CGJRD                                                      *
C    *                                                                   *
C    *  NOTE:  OVER 90 PERCENT OF THE TIME IN STEP 3 IS SPENT IN THIS    *
C    *  SUBROUTINE - IT IS RECOMMEND THAT THIS SUBROUTINE BE REWRITTEN   *
C    *  IN ASSEMBLY LANGUAGE TO OPTIMIZE THE CODE.                       *
C    *                                                                   *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      REAL*8 KEMAT, G , P
      INTEGER*2 GNMAT
      DIMENSION GNMAT( 64,72),  KEMAT(3,2628), G(1575), P(1575)
      M=0
      DO 20 I=1,72
      K=GNMAT(INEL,I)
      DO 20 J=I,72
      L=GNMAT(INEL,J)
      M=M+1
      G(K)=G(K)+KEMAT(ITEL,M)*P(L)
      IF(I.EQ.J) GO TO 20
      G(L)=G(L)+KEMAT(ITEL,M)*P(K)
   20 CONTINUE
      RETURN
      END
```

```
C    MAIN PROGRAM STEP 3 (ITERATION FROM DISK VERSION)                     ESD
C                                                                          ESD
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESD
C    *                                                                   * ESD
C    *   STEP 3 PERFORMS FIVE FUNCTIONS                                  * ESD
C    *                                                                   * ESD
C    *   1.     COMBINES NON-ZERO DISPLACEMENT BOUNDARY CONDITIONS WITH  * ESD
C    *          THE FORCE VECTOR                                         * ESD
C    *                                                                   * ESD
C    *   2.     SOLVES THE SYSTEM OF LINEAR EQUATIONS BY MINIMIZING THE  * ESD
C    *          TOTAL POTENTIAL ENERGY                                   * ESD
C    *                                                                   * ESD
C    *   3.     CALCULATES AND PRINTS THE STRAIN ENERGY AND OTHER        * ESD
C    *          CONVERGENCE PARAMETERS                                   * ESD
C    *                                                                   * ESD
C    *   4.     PRINTS THE COORDINATE AND THREE (X,Y,&Z) DISPLACEMENTS   * ESD
C    *          AT EACH NODE                                             * ESD
C    *                                                                   * ESD
C    *   5.     WRITES DISPLACEMENTS ON DISK AND/OR CARDS                * ESD
C    *                                                                   * ESD
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESD
C                                                                          ESD
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ESD
C    *                                                                   * ESD
C    *   VARIABLE DEFINITIONS AND DIMENSIONS FOR STEP 3                  * ESD
C    *                                                                   * ESD
C    *   ACB - - - - - - - -  CHANGE IN STRAIN ENERGY OVER STRAIN ENERGY * ESD
C    *                                                                   * ESD
C    *   B(NGLDF) - - - - - FORCE VECTOR                                 * ESD
C    *                                                                   * ESD
C    *   BNRM - - - - - - - - MAGNITUDE OF THE FORCE VECTOR              * ESD
C    *                                                                   * ESD
C    *   DBC(MM) - - - - -  NON-ZERO DISPLACEMENT BOUNDARY CONDITIONS    * ESD
C    *                                                                   * ESD
C    *   DELE - - - - - - - - CHANGE IN STRAIN ENERGY                    * ESD
C    *                                                                   * ESD
C    *   DFLXNR - - - - - - - MAGNITUDE OF THE CHANGE IN DISPLACEMENT    * ESD
C    *                        VECTOR                                     * ESD
C    *                                                                   * ESD
C    *   ENGY1 - - - - - - -  STRAIN ENERGY                              * ESD
C    *                                                                   * ESD
C    *   ENGY2 - - - - - - -  STRAIN ENERGY                              * ESD
C    *                                                                   * ESD
C    *   EPS - - - - - - - -  ENERGY CONVERGENCE TEST PARAMETER          * ESD
C    *                                                                   * ESD
C    *   G(NGLDF) - - - - - RESULTING VECTOR FROM KX OR KP MATRIX-       * ESD
C    *                        VECTOR PRODUCT                             * ESD
C    *                                                                   * ESD
C    *   GNMAT(NEL,72) - -  RELATES LOCAL AND GLOBAL DEGREES-OF-FREEDOM  * ESD
C    *                                                                   * ESD
C    *   ICRD - - - - - - - - UNIT NUMBER FOR CARD READER                * ESD
```

```
C     *                                                                  * ESD  520
C     *   IUDA - - - - - - - - UNIT NUMBER FOR DIRECT ACCESS FILE         * ESD  530
C     *                                                                  * ESD  540
C     *   IX26(NEL) - - - -    ELEMENT TYPE FOR EACH ELEMENT               * ESD  550
C     *                                                                  * ESD  560
C     *   KEMAT(2628) - - -    UPPER SYMMETRIC PORTION OF AN ELEMENT       * ESD  570
C     *                        STIFFNESS MATRIX STORED AS A ONE DIMENSIONAL* ESD  580
C     *                        ARRAY                                       * ESD  590
C     *                                                                  * ESD  600
C     *   KOUNT - - - - - - -  TOTAL NUMBER OF ITERATIONS                  * ESD  610
C     *                                                                  * ESD  620
C     *   LDBC(NDBC) - - - - INDEX FOR DISPLACEMENT BOUNDARY CONDITIONS    * ESD  630
C     *                                                                  * ESD  640
C     *   LIST - - - - - - - - UNIT NUMBER FOR PRINTER                     * ESD  650
C     *                                                                  * ESD  660
C     *   LLDBC(MM) - - - -    INDEX NUMBERS FOR NON-ZERO DISPLACEMENT     * ESD  670
C     *                        BOUNDARY CONDITIONS                         * ESD  680
C     *                                                                  * ESD  690
C     *   MM - - - - - - - - - NUMBER OF NON-ZERO DISPLACEMENT BOUNDARY    * ESD  700
C     *                        CONDITIONS                                  * ESD  710
C     *                                                                  * ESD  720
C     *   NDBC - - - - - - - - NUMBER OF DISPLACEMENT BOUNDARY CONDITIONS  * ESD  730
C     *                                                                  * ESD  740
C     *   NEL - - - - - - - -  NUMBER OF ELEMENTS                          * ESD  750
C     *                                                                  * ESD  760
C     *   NGLDF - - - - - - -  NUMBER OF DEGREES-OF-FREEDOM (GLOBAL SYSTEM)* ESD  770
C     *                                                                  * ESD  780
C     *   NTYEL - - - - - - -  NUMBER OF UNIQUE ELEMENTS                   * ESD  790
C     *                                                                  * ESD  800
C     *   P(NGLDF) - - - - - - CORRECTION VECTOR IN CONJUGATE GRADIENT     * ESD  810
C     *                        ROUTINE                                     * ESD  820
C     *                                                                  * ESD  830
C     *   R(NGLDF) - - - - - - RESIDUE VECTOR IN CONJUGATE GRADIENT        * ESD  840
C     *                        ROUTINE                                     * ESD  850
C     *                                                                  * ESD  860
C     *   RNRM - - - - - - - - MAGNITUDE OF THE RESIDUE VECTOR             * ESD  870
C     *                                                                  * ESD  880
C     *   UX(NGNP) - - - - - - DISPLACEMENTS IN THE X-DIRECTION            * ESD  890
C     *                                                                  * ESD  900
C     *   UY(NGNP) - - - - - - DISPLACEMENTS IN THE Y-DIRECTION            * ESD  910
C     *                                                                  * ESD  920
C     *   UZ(NGNP) - - - - - - DISPLACEMENTS IN THE Z-DIRECTION            * ESD  930
C     *                                                                  * ESD  940
C     *   X(NGLDF) - - - - - - DISPLACEMENT VECTOR                         * ESD  950
C     *                                                                  * ESD  960
C     *   XCORD(NGNP) - - -    X-COORDINATE (GLOBAL SYSTEM)                * ESD  970
C     *                                                                  * ESD  980
C     *   XI(NGNP) - - - - - - R-COORDINATE (GLOBAL SYSTEM)                * ESD  990
C     *                                                                  * ESD 1000
C     *   YCORD(NGNP) - - -    Y-COORDINATE (GLOBAL SYSTEM)                * ESD 1010
C     *                                                                  * ESD 1020
```

```
C   *  YT(NGNP) - - - - - THETA-COORDINATE (GLOBAL SYSTEM)                 *
C   *                                                                      *
C   *  ZCORD(NGNP) - - -  Z-COORDINATE (GLOBAL SYSTEM)                     *
C   *                                                                      *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                      *
C   *  LOGICAL SWITCH INFORMATION ( ALL SWITCHES INITIALLY .FALSE.)        *
C
C   *  SW(1)  .TRUE.     CONVERGED TO A SOLUTION                           *
C   *         .FALSE.    DID NOT CONVERGE TO A SOLUTION                    *
C   *         SET IN     CGJRD                                             *
C   *         TESTED IN MAIN                                               *
C   *                                                                      *
C   *  SW(2)  .TRUE.     CHANGE IN STRAIN ENERGY/STRAIN ENERGY .LT. EPS*
C   *         .FALSE.    CHANGE IN STRAIN ENERGY/STRAIN ENERGY .GE. EPS*
C   *         SET IN     CGJRD                                             *
C   *         TESTED IN CGJRD                                              *
C   *                                                                      *
C   *  SW(3)  .TRUE.     EXCEEDED MAXIMUM NUMBER OF ITERATIONS             *
C   *         .FALSE.    DID NOT EXCEED MAXIMUM NUMBER OF ITERATIONS       *
C   *         SET IN     CGJRD                                             *
C   *         TESTED IN CGJRD                                              *
C   *                                                                      *
C   *  SW(4)  .TRUE.     MAGNITUDE OF THE RESIDUE VECTOR .LT. 1.00         *
C   *         .FALSE.    MAGNITUDE OF THE RESIDUE VECTOR .GE. 1.00         *
C   *         SET IN     CGJRD                                             *
C   *         TESTED IN CGJRD                                              *
C   *                                                                      *
C   *  SW(8)  .TRUE.     INITIAL DISPLACEMENTS READ FROM DISK              *
C   *         .FALSE.    INITIAL DISPLACEMENTS READ FROM CARDS             *
C   *         SET IN     MAIN                                              *
C   *         TESTED IN CGJRD                                              *
C   *  SW(5), SW(6), SW(7), SW(9), SW(10), SW(11) & SW(12) NOT USED        *
C   *                                                                      *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT  REAL*8  (A-H,O-Z)
      REAL*8KEMAT
      LOGICAL*1   SW
      INTEGER*2   GNMAT,LDBC,LLDBC, IX26
      COMMON / KEGN /   KEMAT(  2628), GNMAT(120,72),IX26(121)
      COMMON / CGVECT / G(2856),X(2856),B(2856),P(2856),R(2856)
      COMMON /BC/  DBC(200),  LLDBC(200),  LDBC(700),  NDBC
      COMMON /SCALAR/   EPS, ACB, XNRM, BNRM, RNRM,DELXNR,
     1           DELE, ENGY1, ENGY2,
     2       NGLDF, LIMIT, NLL, KOUNT, ITYEL, ITRLMT, MM
      COMMON / HEAD /  HED(10),ICRD,LIST,IPAGE,LINE
      COMMON /SWITCH /      SW(12)
C
```

```
C   CAUTION:  DO NOT DIMENSION UX GREATER THAN 1314 - SEE EQUIVALENCE
C             STATEMENT BELOW
C
      DIMENSION  XCORD(952), YCORD(952), ZCORD(952),
     1          UX(952), UY(952), UZ(952)
      EQUIVALENCE  (XCORD(1),G(1)),  (YCORD(1) ,B(1)),  (ZCORD(1), P(1))
      EQUIVALENCE (XEMAT(1), UX(1)) ,(KEMAT(1315), UY(1)) ,
     1          (R(1), UZ(1))
  100 FORMAT(2I5,2F10.0, I5 )
  102 FORMAT(5Z16)
  201 FORMAT('OMAXIMUM NUMBER OF ITERATIONS FOR THIS RUN IS' , I5)
  301 FORMAT('ODELTA STRAIN ENERGY / STRAIN ENERGY DID CONVERGE TO '
     1     , E14.7,' AFTER' , I6, ' ITERATIONS AND ACCURATE TO',E14.7)
  302 FORMAT(' TIME IN CG SUBROUTINE IS ' , F7.2, ' SECONDS')
  303 FORMAT('ODELTA STRAIN ENERGY / STRAIN ENERGY DID NOT CONVERGE TO '
     1     , E14.7,' AFTER' , I6, ' ITERATIONS BUT ACCURATE TO',E14.7)
  304 FORMAT('0', G14.7, ' IS THE INITIAL GUESS FOR ALL DISPLACEMENTS')
  305 FORMAT ('OTHE INITIAL GUESSES FOR DISPLACEMENTS ARE READ FROM CARD
     1S AND MULTIPLIED BY A FACTOR OF' , G14.7)
  306 FORMAT ('OTHE INITIAL GUESSES FOR DISPLACEMENTS ARE READ FROM CARD
     1S AND THE Z-DISPLACEMENTS ARE MULTIPLIED BY A FACTOR OF', G14.7)
  307 FORMAT('OTHE INITIAL GUESSES FOR DISPLACEMENTS ARE READ FROM DISK'
     1    )
  308 FORMAT('OSTRAIN ENERGY',   T50, E14.7 ,' K IN-LBS' /
     1         ' CHANGE IN STRAIN ENERGY',   T50,  E14.7 ,' K IN-LBS' /
     2         ' MAGNITUDE OF THE RESIDUE VECTOR',  T50, E14.7 ,' K LBS'/
     3         ' MAGNITUDE OF THE FORCE VECTOR',   T50, E14.7 ,' K LBS'/
     4         ' MAGNITUDE OF THE CHANGE IN DISPLACEMENT VECTOR',T50,E14.7
     5,' IN'/' MAGNITUDE OF THE DISPLACEMENT VECTOR',   T50, E14.7,' IN')
 2000 FORMAT('0',7X,'NODE',13X,'X-COORD',13X, 'Y-COORD',13X, 'Z-COORD',
     1          9X, 'X-DISPL', 13X, 'Y-DISPL',13X, 'Z-DISPL' /
     2       27X, 'INS', 17X, 'INS',17X, 'INS', 13X, 'INS', 17X, 'INS',
     3         17X, 'INS' / )
 2001 FORMAT(6X, I5, 3F20.5, 3E20.7)
      DEFINE FILE 3(55,6500,U,IDXDA)
      IUDA = 3
      ICRD = 5
      LIST = 6
C
C   READ CARD DATA FOR STEP 3
C
      READ(ICRD,100)INTXMD,ITRLMT, FACTOR, EPS, IDSPL
C
C   READ DATA GENERATED IN STEP 2
C
      READ (IUDA'1)     NEL, NGLDF,      NDBC, NTYEL, LIMIT, NGNP,NMTL,
     1         HED, IPAGE, AMBTMP
      READ (IUDA'4)     (  B(J),J=1,NGLDF)
     1    , SW, NOCNV,    (LDBC(J),J=1,NDBC)
      READ (IUDA'5)     (  P(J),J=1,NGLDF)
     1                                    , (IX26(J),J=1,NEL)
```

```
      READ (IUDA'6)    ((GNMAT(I,J),J=1,72),I=1,NEL)
      IDXUA = NTYEL+7
      CALL TITLE
      WRITE(LIST,201)  ITRLMT
C
C   DETERMINE INITIAL GUESS FOR THE DISPLACEMENT VECTOR
C
      GO TO ( 21,22,23,24), INTXMD
   21 DO 50 I=1,NGLDF
   50 X(I) = FACTOR
      WRITE(LIST,304)  FACTOR
      GO TO 29
   22 READ(5,102) (X(J),J=1,NGLDF)
      DO 61 I=1,NGLDF
   61 X(I) = FACTOR*X(I)
      WRITE(LIST,305)  FACTOR
      GO TO 29
   23 READ(5,102)  (X(J),J=1,NGLDF)
      DO 10 I=1,NGNP
   10 X(3*I) = FACTOR*X(3*I)
      WRITE(LIST,306)  FACTOR
      GO TO 29
   24 READ (IUDA'IDXDA)SW,MM, (DBC(J),J=1,MM), (LLDBC(J),J=1,MM), KOUNT,
     1          (X(J),J=1,NGLDF)
      SW(8) = .TRUE.
      SW(3) = .FALSE.
   35 IF( .NOT. SW(10) )  GO TO 33
      MM = 0
   33 READ (IUDA'IDXDA)   (B(J),J=1,NGLDF)
      WRITE(LIST,307)
      GO TO 34
   29 CONTINUE
C
C   CONVERT INITIAL DISPLACEMENT BOUNDARY CONDITIONS TO FORCE BOUNDARY
C   CONDITIONS
C
      MM = 0
      DO 60 I=1,NGLDF
      IF( DABS(P(I)) .LT. 1.D-14)  GO TO 60
      MM = MM + 1
      DBC(MM) = P(I)
      LLDBC(MM) = I
   60 CONTINUE
      DO 13 I=1,NGLDF
   13 G(I) = 0.D0
      DO 15 INEL=1,NEL
      LOE = IX26(INEL) + 6
      READ(3'LOE)  KEMAT
   15 CALL GVT(GNMAT, KEMAT, G, P, INEL)
      DO 80 I=1,NGLDF
   80 B(I) = B(I) - G(I)
```

```
   34 CALL TIMON                                                          ESD 2560
C                                                                         ESD 2570
C   MINIMIZE TOTAL POTENTIAL ENERGY                                       ESD 2580
C                                                                         ESD 2590
      CALL CGJRD                                                          ESD 2600
      CALL TIMECK(ITIME)                                                  ESD 2610
      SEC = ITIME/100.                                                    ESD 2620
C                                                                         ESD 2630
C   WRITE FORCE AND DISPLACEMENT DATA ON DISK FOR USE IN FUTURE RUNS      ESD 2640
C                                                                         ESD 2650
      IF( MM .NE. 0 )  GO TO 31                                           ESD 2660
      SW(10) = .TRUE.                                                     ESD 2670
      MM = 1                                                              ESD 2680
      DBC(1) = CDO                                                        ESD 2690
   31 IDXDA = NTYEL + 7                                                   ESD 2700
      WRITE(IUDA'IDXDA)SW,MM, (DBC(J),J=1,MM), (LLDBC(J),J=1,MM), KOUNT,ESD 2710
     1          (X(J),J=1,NGLDF)                                          ESD 2720
      WRITE(IUDA'IDXDA)    (B(J),J=1,NGLDF)                               ESD 2730
C                                                                         ESD 2740
C   PRINT STRAIN ENERGY AND CONVERGENCE DATA                              ESD 2750
C                                                                         ESD 2760
      IF(SW(1)) GO TO 20                                                  ESD 2770
      WRITE(LIST,303)  EPS, KOUNT, ACB                                    ESD 2780
      GO TO 9                                                             ESD 2790
   20 WRITE(LIST,301)  EPS, KOUNT, ACB                                    ESD 2800
    9 WRITE(LIST,302)  SEC                                                ESD 2810
      WRITE(LIST,308)  ENGY1, DELE, RNRM, BNRM, DELXNR, XNRM              ESD 2820
      IF(IDSPL .EQ. 0)  GO TO 32                                          ESD 2830
      IF(IDSPL .EQ. 2)  GO TO 43                                          ESD 2840
C                                                                         ESD 2850
C   PRINT AND/OR PUNCH THE DISPLACEMENT VECTOR                            ESD 2860
C                                                                         ESD 2870
      READ(IUDA'3)(XCORD(J),J=1,NGNP), (YCORD(J),J=1,NGNP),               ESD 2880
     1             (ZCORD(J),J=1,NGNP)                                    ESD 2890
      CALL TITLE                                                          ESD 2900
      WRITE(LIST,2000)                                                    ESD 2910
      DO 30 I=1,NGNP                                                      ESD 2920
      IF(LINE .LT. 48) GO TO 94                                           ESD 2930
      CALL TITLE                                                          ESD 2940
      WRITE(LIST,2000)                                                    ESD 2950
   94 CONTINUE                                                            ESD 2960
      LINE = LINE + 1                                                     ESD 2970
      UX(I) = X(3*I-2)                                                    ESD 2980
      UY(I) = X(3*I-1)                                                    ESD 2990
      UZ(I) = X(3*I)                                                      ESD 3000
   30 WRITE(LIST,2001)  I,  XCORD(I) , YCORD(I), ZCORD(I),                ESD 3010
     1             UX(I), UY(I), UZ(I)                                    ESD 3020
      IF(IDSPL .EQ. 1)  GO TO 32                                          ESD 3030
   43 WRITE(7,102)  (X(I),I=1,NGLDF)                                      ESD 3040
   32 STOP                                                                ESD 3050
      END                                                                 ESD 3060
```

```
      SUBROUTINE  CGJRD
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                               *
C   *  SUBROUTINE CGJRD MINIMIZES THE TOTAL POTENTIAL ENERGY BY THE  *
C   *  CONJUGATE GRADIENT METHOD AND TESTS THE STRAIN ENERGY FOR    *
C   *  CONVERGENCE                                                  *
C   *                                                               *
C   *  THIS SUBROUTINE IS CALLED BY -                               *
C   *        MAIN                                                   *
C   *                                                               *
C   *  THIS SUBROUTINE CALLS                                        *
C   *        GVT                                                    *
C   *                                                               *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      IMPLICIT  REAL*8  (A-H,O-Z)
      REAL*8KEMAT
      INTEGER*2   GNMAT,LDBC,LLDBC, IX26
      LOGICAL*1   SW
      COMMON / KEGN /   KEMAT(  2628), GNMAT(120,72),IX26(121)
      COMMON / CGVECT / G(2856),X(2856),B(2856),P(2856),R(2856)
      COMMON /SCALAR/   EPS, ACB, XNRM, BNRM, RNRM,DELXNR,
     1           DELE, ENGY1, ENGY2,
     2        NGLDF, LIMIT, NEL, KOUNT, ITYEL, ITRLMT, MM
      COMMON /BC/  DBC(200),  LLDBC(200),  LDBC(700),   NDBC
      COMMON /SWITCH /       SW(12)
      ITKNT = 0
      IF(SW(8) )  GO TO 2
      KOUNT = 0
      BETA = 0.D0
C
C  RESTART ITERATIVE PROCESS BY FINDING NEW RESIDUE VECTOR
C
    2 DO 10 I = 1,NGLDF
   10 G(I) = 0.D0
      DO 70 I=1,NDBC
      B(LDBC(I)) = 0.D0
   70 X(LDBC(I)) = 0.D0
      DO 31 INEL=1,NEL
      LQE = IX26(INEL) + 6
      READ(3'LQE)  KEMAT
   31 CALL GVT(GNMAT, KEMAT, G, X, INEL)
      DO 32 I=1,NDBC
   32 G(LDBC(I)) = 0.D0
      RNRM1 = 0.D0
      DO 20 I = 1,NGLDF
      P(I) = B(I) - G(I)
      R(I) = P(I)
   20 RNRM1 = RNRM1 + R(I) * R(I)
```

```
C
C   CALCULATE STRAIN ENERGY
C
      IF( .NOT. (SW(3) .OR. SW(4))) GO TO 5
      IF(MM .EQ. 0) GO TO 95
      DO 90 I=1,MM
   90 X(LLDBC(I)) = DBC(I)
   95 ENGY1 = 0.D0
      DO 17 I=1,NGLDF
   17 G(I) = 0.D0
      DO 34 INEL=1,NEL
      LOE = IX26(INEL) + 6
      READ(3'LOE)  KEMAT
   34 CALL GVT(GNMAT, KEMAT, G, X, INEL)
      DO 92 I=1,NGLDF
   92 ENGY1 = ENGY1 + X(I)   * G(I)
      ENGY1 = .5D0 * ENGY1
      ENGY2 = ENGY1
C
C   FIND ALFA
C
    5 DO 35 I=1,NGLDF
   35 G(I) = 0.D0
      DO 33 INEL=1,NEL
      LOE = IX26(INEL) + 6
      READ(3'LOE)  KEMAT
   33 CALL GVT(GNMAT, KEMAT, G, P, INEL)
      DO 36 I=1,NDBC
   36 G(LDBC(I)) = 0.D0
      PKP = 0.D0
      DO 30 I = 1,NGLDF
   30 PKP = PKP + P(I) * G(I)
      ALFA = RNRM1 / PKP
      IF(SW(3)) GO TO 25
C
C   CORRECT THE DISPLACEMENT VECTOR
C
      DO 40 I = 1,NGLDF
   40 X(I) = X(I) + ALFA * P(I)
      ITKNT = ITKNT + 1
      KOUNT = KOUNT + 1
      IF(ITKNT .LT. ITRLMT) GO TO 7
      SW(3) = .TRUE.
      GO TO 2
    7 IF(SW(4)) GO TO 12
      IF (RNRM1 .GT. 1.) GO TO 11
      SW(4) = .TRUE.
      GO TO 2
   11 RNRM2 = 0.D0
C
C   FIND NEW RESIDUE VECTOR AND NEW P VECTOR
```

```
C
      DO 50 I = 1,NGLDF
      R(I) = R(I) - ALFA*G(I)
   50 RNRM2 = RNRM2 + R(I) * R(I)
      BETA = RNRM2/RNRM1
      DO 60  I = 1,NGLDF
   60 P(I) =  R(I) + BETA * P(I)
      RNRM1 = RNRM2
      GO TO 5
C
C   FIND CHANGE IN STRAIN ENERGY FROM ALFA & RNRM1
C
   12 DELE = .5D0 * ALFA * RNRM1
      ENGY2 = ENGY2 - DELE
      IF(DELE/ENGY2 .LT. EPS)  GO TO 14
      SW(2) = .FALSE.
      GO TO 11
   14 IF(SW(2)) GO TO 15
      SW(2) = .TRUE.
      GO TO 2
   15 SW(1) = .TRUE.
C
C   CALCULATE CONVERGENCE PARAMETERS
C
   25 XNRM = 0.D0
      BNRM = 0.D0
      DELXNR = 0.D0
      DO 16 I=1,NGLDF
      XNRM = XNRM + X(I) * X(I)
      BNRM = BNRM + B(I) * B(I)
   16 DELXNR = DELXNR + P(I) * P(I)
      XNRM = DSQRT(XNRM)
      BNRM = DSQRT(BNRM)
      RNRM = DSQRT(RNRM1)
      DELXNR = ALFA * DSQRT(DELXNR)
      IF( .NOT. SW(2))  DELE = .5D0 * ALFA * RNRM1
      ACF = DELE / ENGY2
      RETURN
      END
```

```
      SUBROUTINE TITLE
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   *  SUBROUTINE TITLE PRINTS THE HEADING ON EACH PAGE               *
C   *  THIS SUBROUTINE IS CALLED BY -                                 *
C   *        MAIN                                                     *
C   *                                                                 *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / HEAD /  HED(10),ICRD,LIST,IPAGE,LINE
  100 FORMAT (1H1,'FEM 72-DOF GENERAL HEXAHEDRONS THERMO-ELASTIC, VARYIN
     1G MATERIAL PROPERTIES, DANA', 9X, 'PAGE', I3)
  101 FORMAT (1H0,10A8 )
      LIST = 6
      IWRT = 6
      WRITE (LIST,100) IPAGE
      WRITE (LIST,101) HED
      IPAGE= IPAGE +1
      LINE = 0
      RETURN
      END
```

```
      SUBROUTINE GVT(GNMAT,KEMAT,G,P,INEL)
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                               *
C   *  SUBROUTINE GVT FORMS THE MATRIX-VECTOR PRODUCT KP=G WHERE K   *
C   *  REPRESENTS THE NON-FORMED GLOBAL STIFFNESS MATRIX             *
C   *                                                               *
C   *  THIS SUBROUTINE IS CALLED BY -                                *
C   *        MAIN                                                   *
C   *        CGJRD                                                  *
C   *                                                               *
C   *  NOTE:  OVER 90 PERCENT OF THE TIME IN STEP 3 IS SPENT IN THIS *
C   *  SUBROUTINE - IT IS RECOMMEND THAT THIS SUBROUTINE BE REWRITTEN *
C   *  IN ASSEMBLY LANGUAGE TO OPTIMIZE THE CODE.                    *
C   *                                                               *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      REAL*8 KEMAT, G , P
      INTEGER*2 GNMAT
      DIMENSION GNMAT(120,72), KEMAT(2628), G(2856), P(2856)
      M=0
      DO 20 I=1,72
      K=GNMAT(INEL,I)
      DO 20 J=1,72
      L=GNMAT(INEL,J)
      M=M+1
      G(K)=G(K)+KEMAT(M)*P(L)
      IF(I.EQ.J) GO TO 20
      G(L)=G(L)+KEMAT(M)*P(K)
   20 CONTINUE
      RETURN
      END
```

```
C     MAIN PROGRAM STEP 4
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *  STEP 4 PERFORMS THREE FUNCTIONS                                  *
C     *                                                                   *
C     *  1.     READ PROBLEM DATA AND DISPLACEMENTS FROM DISK             *
C     *                                                                   *
C     *  2.     CALCULATES SIX STRESS COMPONENTS AT EACH NODE FOR EACH    *
C     *         ELEMENT IN RECTANGULAR OR CYLINDRICAL COORDINATES         *
C     *                                                                   *
C     *  3.     PRINTS STRESSES AND/OR CYLINDRICAL DISPLACEMENTS          *
C     *                                                                   *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                   *
C     *  VARIABLE DEFINITIONS AND DIMENSIONS FOR STEP 4                   *
C     *                                                                   *
C     *  ALFA1(NMTL) - - -  THERMAL EXPANSION COEFFICIENT 11              *
C     *                                                                   *
C     *  ALFA2(NMTL) - - -  THERMAL EXPANSION COEFFICIENT 22              *
C     *                                                                   *
C     *  ALFA3(NMTL) - - -  THERMAL EXPANSION COEFFICIENT 33              *
C     *                                                                   *
C     *  AMBTMP - - - - - - INITIAL TEMPERATURE                           *
C     *                                                                   *
C     *  D(6,6) - - - - - - ELASTIC MATRIX                                *
C     *                                                                   *
C     *  E(NMTL,9,NTMP) - - MATERIAL PROPERTIES                           *
C     *                                                                   *
C     *  ETM(9) - - - - - - NINE INDEPENDENT ORTHOTROPIC MATERIAL         *
C     *                     CONSTANTS                                     *
C     *                                                                   *
C     *  FIBORT(NMTL) - - - DIRECTION OF PRINCIPAL AXIS FOR EACH          *
C     *                     MATERIAL                                      *
C     *                                                                   *
C     *  ICRD - - - - - - - UNIT NUMBER FOR CARD READER                   *
C     *                                                                   *
C     *  ISTRS - - - - - -  CODE TO SPECIFY STRESSES IN RECTANGULAR OR    *
C     *                     CYLINDRICAL COORDINATES                       *
C     *                                                                   *
C     *  IUDA - - - - - - - UNIT NUMBER FOR DIRECT ACCESS FILE            *
C     *                                                                   *
C     *  IX(NEL,27) - - - - RELATES LOCAL AND GLOBAL NODAL POINTS         *
C     *                                                                   *
C     *  LIST - - - - - - - UNIT NUMBER FOR PRINTER                       *
C     *                                                                   *
C     *  MTLND(NGNP) - - -  MATERIAL AT EACH NODE                         *
C     *                                                                   *
C     *  NEL - - - - - - -  NUMBER OF ELEMENTS                            *
C     *                                                                   *
```

```
C    *   NGLDF - - - - - -  NUMBER OF DEGREES-OF-FREEDOM (GLOBAL SYSTEM)*
C    *                                                                   *
C    *   NMTL - - - - - - - NUMBER OF MATERIALS                          *
C    *                                                                   *
C    *   NTMP(NMTL) - - - - NUMBER OF MATERIAL PROPERTIES SPECIFIED FOR  *
C    *                      EACH ELEMENT                                 *
C    *                                                                   *
C    *   TMPEL(NMTL,NTMP) - TEMPERATURES AT WHICH MATERIAL PROPERTIES    *
C    *                      ARE SPECIFIED FOR EACH MATERIAL              *
C    *                                                                   *
C    *   TMPND(NGNP) - - -  FINAL NODAL POINT TEMPERATURES               *
C    *                                                                   *
C    *   U(NGLNF) - - - - - DISPLACEMENT VECTOR                          *
C    *                                                                   *
C    *   UX(NGNP) - - - - - X-DIRECTION DISPLACEMENTS                    *
C    *                                                                   *
C    *   UY(NGNP) - - - - - Y-DIRECTION DISPLACEMENTS                    *
C    *                                                                   *
C    *   UZ(NGNP) - - - - - Z-DIRECTION DISPLACEMENTS                    *
C    *                                                                   *
C    *   X(NGNP) - - - - -  X-COORDINATE (GLOBAL SYSTEM)                 *
C    *                                                                   *
C    *   Y(NGNP) - - - - -  Y-COORDINATE (GLOBAL SYSTEM)                 *
C    *                                                                   *
C    *   Z(NGNP) - - - - -  Z-COORDINATE (GLOBAL SYSTEM)                 *
C    *                                                                   *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL*1   SW(12)
      INTEGER*2   IX, MTLND, LLDBC(200)
      COMMON / GENLS /  NEL, NGLDF, NGNP, NMTL, INEL, ILNP, IGNP, IMTL
      COMMON /MATL /    E(9,9,10),D(6,6), FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9),AMBTMP   ,ETM(9), TMPEL(9,10), NTMP(9)
      COMMON  / NODAL /  X(1015), Y(1015), Z(1015), UX(1015),UY(1015),
     1   UZ(1015), TMPND(1015), U(3045), ALFTMP(6), XT(1015), YT(1015),
     2      ISTRS, IX(144,27), MTLND(1015)
      COMMON / HEAD /  HED(10),ICRD,LIST,IPAGE,LINE
  100 FORMAT(I5)
 2000 FORMAT('0',7X,'NODE',13X,'R-COORD',11X, 'THETA-COORD',11X,
     1   'Z-COORD',8X,'RAD-DISPL', 12X,'TANG-DISPL', 11X, 'Z-DISPL'  /
     2   27X, 'INS', 15X, 'DEGREES', 15X, 'INS', 13X, 'INS', 17X,
     3        'DEGREES', 14X, 'INS' /)
 2001 FORMAT(6X, I5, 3F20.5, 3E20.7)
      DIMENSION    DBC( 200)
      DEFINE FILE 3(55,6500,U,IDXDA)
      LIST = 6
      ICRD = 5
      IUDA = 3
C
C   READ STRESS TYPE CODE
```

```
C
      READ(ICRD,100)  ISTRS
C
C   READ DATA GENERATED IN STEP 2 - PROBLEM DATA
C
      READ (IUDA'1)      NEL, NGLDF,       NDBC, NTYEL, LIMIT, NGNP,NMTL,
     1        HED, IPAGE, AMBTMP
      READ (IUDA'2) ((NTMP(J), FIBORT(J), ALFA1(J), ALFA2(J), ALFA3(J)),
     1     (TMPEL(J,I), (E(J,L,I),L=1,9), I=1,10), J=1,NMTL)
     2     , ((IX(I,J),J=1,27),I=1,NEL),
     3    (TMPND(J), MTLND(J),J=1,NGNP)
      READ (IUDA'3)    (X(J),J=1,NGNP), (Y(J),J=1,NGNP),(Z(J),J=1,NGNP)
C
C   READ DATA GENERATED IN STEP 3 - DISPLACEMENTS
C
      IDXDA = NTYEL+7
      READ (IUDA'IDXDA)SW;MM, (DBC(J),J=1,MM), (LLDBC(J),J=1,MM), KOUNT,
     1          (U(J),J=1,NGLDF)
      IF(ISTRS .EQ. 0) GO TO 12
C
C   TRANSFORM X AND Y RECTANGULAR COORDINATES TO R AND THETA
C   CYLINDRICAL COORDINATES
C
      DO 3 I=1,NGNP
      XT(I) = DSQRT(X(I)*X(I) + Y(I)*Y(I) )
    3 YT(I) = 180.D0 * DATAN2(Y(I), X(I) )  /  3.14159265358979D0
      IF(ISTRS .EQ. 1) GO TO 12
C
C   PRINT DISPLACEMENTS IN CYLINDRICAL COORDINATES
C
      CALL TITLE
      WRITE(LIST,2000)
      DO 30 I=1,NGNP
      IF(LINE .LT. 48) GO TO 94
      CALL TITLE
      WRITE(LIST,2000)
   94 CONTINUE
      LINE = LINE + 1
      UX(I) = U(3*I-2) * DCOS(3.14159265358979D0 * YT(I) / 180.D0)
     1      + U(3*I-1) * DSIN(3.14159265358979D0 * YT(I) / 180.D0)
      UY(I) =-U(3*I-2) * DSIN(3.14159265358979D0 * YT(I) / 180.D0)
     1      + U(3*I-1) * DCOS(3.14159265358979D0 * YT(I) / 180.D0)
      UZ(I) = U(3*I)
   30 WRITE(LIST,2001)  I,  XT(I), YT(I), Z(I), UX(I), UY(I), UZ(I)
      IF(ISTRS .EQ. 2) GO TO 9
C
C   CALCULATE STRESSES
C
   12 CALL STRESS
    9 STOP
      END
```

```
      SUBROUTINE TITLE                                                  TI4
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TI4
C     *                                                               * TI4
C     *  SUBROUTINE TITLE PRINTS THE HEADING ON EACH PAGE              * TI4
C     *                                                               * TI4
C     *  THIS SUBROUTINE IS CALLED BY -                                * TI4
C     *        STRESS                                                 * TI4
C     *                                                               * TI4
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TI4
C                                                                       TI4
      IMPLICIT REAL*8 (A-H,O-Z)                                         TI4
      COMMON / HEAD /  HED(10),ICRD,LIST,IPAGE,LINE                     TI4
  100 FORMAT (1H1,'FEM 72-DOF GENERAL HEXAHEDRONS THERMO-ELASTIC, VARYINTI4
     1G MATERIAL PROPERTIES, DANA', 9X, 'PAGE', I3)                     TI4
  101 FORMAT (1H0,10A8 )                                                TI4
      WRITE (LIST,100) IPAGE                                            TI4
      WRITE (LIST,101) HED                                              TI4
      IPAGE= IPAGE +1                                                   TI4
      LINE = 0                                                          TI4
      RETURN                                                            TI4
      END                                                               TI4
```

```
      SUBROUTINE STRESS
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE STRESS CALCULATES AND PRINTS THE SIX STRESS           *
C   *  COMPONENTS AT EACH NODE FOR EACH ELEMENT                         *
C   *                                                                   *
C   *  THIS SUBROUTINE IS CALLED BY -                                   *
C   *        MAIN                                                       *
C   *                                                                   *
C   *  THIS SUBROUTINE CALLS -                                          *
C   *        TITLE                                                      *
C   *        ELASTR                                                     *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
     .INTEGER*2   IX, MTLND
      COMMON / GENLS /  NLL, NGLDF, NGNP, NMTL, INEL, ILNP, IGNP, IMTL
      COMMON /MATL /     E(9,9,10),D(6,6), FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9),AMBTMP    ,ETM(9), TMPEL(9,10), NTMP(9)
      COMMON  / NODAL /  X(1015), Y(1015), Z(1015), UX(1015),UY(1015),
     1   UZ(1015), TMPND(1015), U(3045), ALFTMP(6), XT(1015), YT(1015),
     2      ISTRS, IX(144,27), MTLND(1015)
      COMMON / HEAD /  HED(10),ICRD,LIST,IPAGE,LINE
      DIMENSION  XYZ(24,3), C(3,24), DJ(3,3), DJI(3,3), BA(6,72),
     1      BDB(72,72), SIG(6), UEL(72), SIGSAV(6,24), T(3,3),
     2      SIGRTH(3,3), SIGXY(3,3)
 2000 FORMAT (1H0,9X,6HEL.NO.,6X,4HNODE,9X,8HX-STRESS,6X,8HY-STRESS,
     16X, 8HZ-STRESS,6X, 9HXY-STRESS,5X,9HXZ-STRESS,5X,9HYZ-STRESS/
     2       37X, 'KSI', 10X, 'KSI', 12X, 'KSI', 11X, 'KSI', 11X, 'KSI',
     3       11X, 'KSI'  )
 2001 FORMAT (15X,I11,4X,6D14.5)
 2002 FORMAT (1H0,9X,I5,I11,4X,6D14.5)
 2003 FORMAT('0          EL.NO.      NODE        R-STRESS    THETA-STRES
     1S    Z-STRESS    RTHETA-STRESS   RZ-STRESS    THETAZ-STRESS' /
     2       37X, 'KSI', 10X, 'KSI', 12X, 'KSI', 11X, 'KSI', 11X, 'KSI',
     3       11X, 'KSI'  )
 2004 FORMAT (15X,I11,4X,6D14.5)
 2005 FORMAT (1H0,9X,I5,I11,4X,6D14.5)
C
C SOLVE FOR STRESSES SIG(6) AT EACH NODAL POINTS OF EACH ELEMENT
C
      CALL TITLE
      IF(ISTRS .EQ. 0 .OR. ISTRS .EQ. 4)  GO TO 31
      WRITE(LIST,2003)
      GO TO 32
   31 WRITE (LIST,2000)
   32 CONTINUE
      DO 300 INEL=1,NEL
C
```

```
C PLACE PROPER NODAL DISPLACEMENTS IN U FROM A
C
      DO 121 I=1,24
      UEL (3*I-2) = U(3*IX(INEL,I)-2)
      UEL (3*I-1) = U(3*IX(INEL,I)-1)
  121 UEL (3*I  ) = U(3*IX(INEL,I)  )
C
C FORM NODAL PT. COORDS. MATRIX  XYZ  FOR J(3X3) = C(3X24)*XYZ(24X3)
C
      DO 140 I=1,24
      L = IX(INEL,I)
      XYZ(I,1) = X(L)
      XYZ(I,2) = Y(L)
  140 XYZ(I,3) = Z(L)
C
C CALCULATE BA = D*B, 24 SETS OF (6X72) FOR EACH NODE OF THE ELEMENT
C
      DO 200 ILNP=1,24
      DO 150 I=1,6
  150 SIG(I) = 0.D0
      GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
     1       13,14,15,16,17,18,19,20, 21, 22, 23, 24), ILNP
    1 XSII= -1.D0
      ETAA= -1.D0
      ZTAA= -1.D0
      GO TO 25
    2 XSII= -1.D0/3.D0
      GO TO 25
    3 XSII=  1.D0/3.D0
      GO TO 25
    4 XSII= 1.D0
      GO TO 25
    5 ZTAA= 1.D0
      XSII= -1.D0
      GO TO 25
    6 XSII= -1.D0/3.D0
      GO TO 25
    7 XSII=  1.D0/3.D0
      GO TO 25
    8 XSII=  1.D0
      GO TO 25
    9 ZTAA= -1.D0
      XSII= -1.D0
      ETAA= -1.D0/3.D0
      GO TO 25
   10 ETAA= 1.D0/3.D0
      GO TO 25
   11 XSII= 1.D0
      ETAA= -1.D0/3.D0
      GO TO 25
   12 ETAA=  1.D0/3.D0
```

```
      GO TO 25
   13 ZTAA= 1.D0
      XSII= -1.D0
      ETAA= -1.D0/3.D0
      GO TO 25
   14 ETAA=  1.D0/3.D0
      GO TO 25
   15 XSII= 1.D0
      ETAA= -1.D0/3.D0
      GO TO 25
   16 ETAA=  1.D0/3.D0
      GO TO 25
   17 ETAA=  1.D0
      ZTAA= -1.D0
      XSII= -1.D0
      GO TO 25
   18 XSII= -1.D0/3.D0
      GO TO 25
   19 XSII=  1.D0/3.D0
      GO TO 25
   20 XSII=  1.D0
      GO TO 25
   21 ZTAA=  1.D0
      XSII=- 1.D0
      GO TO 25
   22 XSII= -1.D0/3.D0
      GO TO 25
   23 XSII=  1.D0/3.D0
      GO TO 25
   24 XSII=  1.D0
   25 CONTINUE
C
C     FORM C MATRIX
C
      C(1,1) =(1.D0-ETAA)*(1.D0-ZTAA)*(10.D0+18.D0*XSII-27.D0*XSII**2 -
     1          9.D0*ETAA**2)
      C(1,2) =(1.D0-ETAA)*(1.D0-ZTAA)*(81.D0*XSII**2-18.D0*XSII-27.D0)
      C(1,3) =(1.D0-ETAA)*(1.D0-ZTAA)*(27.D0-18.D0*XSII-81.D0*XSII**2)
      C(1,4) =(1.D0-ETAA)*(1.D0-ZTAA)*(27.D0*XSII**2+9.D0*ETAA**2+
     1          18.D0*XSII-10.D0)
      C(1,5) =(1.D0-ETAA)*(1.D0+ZTAA)*(10.D0+18.D0*XSII-27.D0*XSII**2-
     1          9.D0*ETAA**2)
      C(1,6) =(1.D0-ETAA)*(1.D0+ZTAA)*(81.D0*XSII**2-18.D0*XSII-27.D0)
      C(1,7) =(1.D0-ETAA)*(1.D0+ZTAA)*(27.D0-18.D0*XSII-81.D0*XSII**2)
      C(1,8) =(1.D0-ETAA)*(1.D0+ZTAA)*(27.D0*XSII**2+9.D0*ETAA**2+
     1          18.D0*XSII-10.D0)
      C(1,9) =(1.D0-3.D0*ETAA)*(1.D0-ZTAA)*(9.D0*ETAA**2-9.D0)
      C(1,10)=(1.D0+3.D0*ETAA)*(1.D0-ZTAA)*(9.D0*ETAA**2-9.D0)
      C(1,11)=-C(1,9)
      C(1,12)=-C(1,10)
      C(1,13)=(1.D0-3.D0*ETAA)*(1.D0+ZTAA)*(9.D0*ETAA**2-9.D0)
```

```
      C(1,14)=(1.D0+3.D0*ETAA)*(1.D0+ZTAA)*(9.D0*ETAA**2-9.D0)
      C(1,15)=-C(1,13)
      C(1,16)=-C(1,14)
      C(1,17)=(1.D0+ETAA)*(1.D0-ZTAA)*(10.D0+18.D0*XSII-27.D0*XSII**2-
     1        9.D0*ETAA**2)
      C(1,18)=(1.D0+ETAA)*(1.D0-ZTAA)*(81.D0*XSII**2-18.D0*XSII-27.D0)
      C(1,19)=(1.D0+ETAA)*(1.D0-ZTAA)*(27.D0-18.D0*XSII-81.D0*XSII**2)
      C(1,20)=(1.D0+ETAA)*(1.D0-ZTAA)*(27.D0*XSII**2+9.D0*ETAA**2+
     1        18.D0*XSII-10.D0)
      C(1,21)=(1.D0+ETAA)*(1.D0+ZTAA)*(10.D0+18.D0*XSII-27.D0*XSII**2-
     1        9.D0*ETAA**2)
      C(1,22)=(1.D0+ETAA)*(1.D0+ZTAA)*(81.D0*XSII**2-18.D0*XSII-27.D0)
      C(1,23)=(1.D0+ETAA)*(1.D0+ZTAA)*(27.D0-18.D0*XSII-81.D0*XSII**2)
      C(1,24)=(1.D0+ETAA)*(1.D0+ZTAA)*(27.D0*XSII**2+9.D0*ETAA**2+
     1        18.D0*XSII-10.D0)
      C(2,1) =(1.D0-XSII)*(1.D0-ZTAA)*(10.D0+18.D0*ETAA-9.D0*XSII**2-
     1        27.D0*ETAA**2)
      C(2,2) =(1.D0-3.D0*XSII)*(1.D0-ZTAA)*(9.D0*XSII**2-9.D0)
      C(2,3) =(1.D0+3.D0*XSII)*(1.D0-ZTAA)*(9.D0*XSII**2-9.D0)
      C(2,4) =(1.D0+XSII)*(1.D0-ZTAA)*(10.D0+18.D0*ETAA-9.D0*XSII**2-
     1        27.D0*ETAA**2)
      C(2,5) =(1.D0-XSII)*(1.D0+ZTAA)*(10.D0+18.D0*ETAA-9.D0*XSII**2-
     1        27.D0*ETAA**2)
      C(2,6) =(1.D0-3.D0*XSII)*(1.D0+ZTAA)*(9.D0*XSII**2-9.D0)
      C(2,7) =(1.D0+3.D0*XSII)*(1.D0+ZTAA)*(9.D0*XSII**2-9.D0)
      C(2,8) =(1.D0+XSII)*(1.D0+ZTAA)*(10.D0+18.D0*ETAA-9.D0*XSII**2-
     1        27.D0*ETAA**2)
      C(2,9)=(1.D0-XSII)*(1.D0-ZTAA)*(81.D0*ETAA**2-18.D0*ETAA-27.D0)
      C(2,10)=(1.D0-XSII)*(1.D0-ZTAA)*(27.D0-18.D0*ETAA-81.D0*ETAA**2)
      C(2,11)=(1.D0+XSII)*(1.D0-ZTAA)*(81.D0*ETAA**2-18.D0*ETAA-27.D0)
      C(2,12)=(1.D0+XSII)*(1.D0-ZTAA)*(27.D0-18.D0*ETAA-81.D0*ETAA**2)
      C(2,13)=(1.D0-XSII)*(1.D0+ZTAA)*(81.D0*ETAA**2-18.D0*ETAA-27.D0)
      C(2,14)=(1.D0-XSII)*(1.D0+ZTAA)*(27.D0-18.D0*ETAA-81.D0*ETAA**2)
      C(2,15)=(1.D0+XSII)*(1.D0+ZTAA)*(81.D0*ETAA**2-18.D0*ETAA-27.D0)
      C(2,16)=(1.D0+XSII)*(1.D0+ZTAA)*(27.D0-18.D0*ETAA-81.D0*ETAA**2)
      C(2,17)=(1.D0-XSII)*(1.D0-ZTAA)*(27.D0*ETAA**2+9.D0*XSII**2+
     1        18.D0*ETAA-10.D0)
      C(2,18)=-C(2,2)
      C(2,19)=-C(2,3)
      C(2,20)=(1.D0+XSII)*(1.D0-ZTAA)*(27.D0*ETAA**2+9.D0*XSII**2+18.D0*
     1        ETAA-10.D0)
      C(2,21)=(1.D0-XSII)*(1.D0+ZTAA)*(27.D0*ETAA**2+9.D0*XSII**2+
     1        18.D0*ETAA-10.D0)
      C(2,22)=(1.D0-3.D0*XSII)*(1.D0+ZTAA)*(9.D0-9.D0*XSII**2)
      C(2,23)=(1.D0+3.D0*XSII)*(1.D0+ZTAA)*(9.D0-9.D0*XSII**2)
      C(2,24)=(1.D0+XSII)*(1.D0+ZTAA)*(27.D0*ETAA**2+9.D0*XSII**2+
     1        18.D0*ETAA-10.D0)
      C(3,1) =(1.D0-XSII)*(1.D0-ETAA)*(10.D0-9.D0*XSII**2-9.D0*ETAA**2)
      C(3,2) =(1.D0-3.D0*XSII)*(1.D0-E[illegible])*(9.D0*XSII**2-9.D0)
      C(3,3) =(1.D0+3.D0*XSII)*(1.D0-[illegible](9.D0*XSII**2-9.D0)
      C(3,4) =(1.D0+XSII)*(1.D0-ETAA[illegible]0-9.D0*XSII**2-9.D0*ETAA**2)
```

```
      DO 26 N=1,4
   26 C(3,N+4) = -C(3,N)
      C(3,9) =(1.D0-3.D0*ETAA)*(1.D0-XSII)*(9.D0*ETAA**2-9.D0)
      C(3,10)=(1.D0+3.D0*ETAA)*(1.D0-XSII)*(9.D0*ETAA**2-9.D0)
      C(3,11)=(1.D0-3.D0*ETAA)*(1.D0+XSII)*(9.D0*ETAA**2-9.D0)
      C(3,12)=(1.D0+3.D0*ETAA)*(1.D0+XSII)*(9.D0*ETAA**2-9.D0)
      DO 27 N=9,12
   27 C(3,N+4) = -C(3,N)
      C(3,17)=(1.D0-XSII)*(1.D0+ETAA)*(10.D0-9.D0*XSII**2-9.D0*ETAA**2)
      C(3,18)=(1.D0-3.D0*XSII)*(1.D0+ETAA)*(9.D0*XSII**2-9.D0)
      C(3,19)=(1.D0+3.D0*XSII)*(1.D0+ETAA)*(9.D0*XSII**2-9.D0)
      C(3,20)=(1.D0+XSII)*(1.D0+ETAA)*(10.D0-9.D0*XSII**2-9.D0*ETAA**2)
      DO 28 N=17,20
   28 C(3,N+4) = -C(3,N)
      DO 30 II=1,3
      DO 30 KK=1,3
      DJ(II,KK) = 0.D0
      DO 30 JJ=1,24
   30 DJ(II,KK) = DJ(II,KK) + C(II,JJ) * XYZ(JJ,KK) / 64.D0
C
C FORM INVERSE J MATRIX  DJI(3X3)
C
      DETJ     = DJ(1,1)*(DJ(2,2)*DJ(3,3) - DJ(2,3)*DJ(3,2))
     1          +DJ(1,2)*(DJ(2,3)*DJ(3,1) - DJ(2,1)*DJ(3,3))
     2          +DJ(1,3)*(DJ(3,2)*DJ(2,1) - DJ(3,2)*DJ(3,1))
      DJI(1,1) = (DJ(2,2)*DJ(3,3) - DJ(2,3)*DJ(3,2)) /DETJ
      DJI(1,2) = (DJ(3,2)*DJ(1,3) - DJ(3,3)*DJ(1,2)) /DETJ
      DJI(1,3) = (DJ(1,2)*DJ(2,3) - DJ(1,3)*DJ(2,2)) /DETJ
      DJI(2,1) = (DJ(2,3)*DJ(3,1) - DJ(2,1)*DJ(3,3)) /DETJ
      DJI(2,2) = (DJ(3,3)*DJ(1,1) - DJ(3,1)*DJ(1,3)) /DETJ
      DJI(2,3) = (DJ(1,3)*DJ(2,1) - DJ(1,1)*DJ(2,3)) /DETJ
      DJI(3,1) = (DJ(2,1)*DJ(3,2) - DJ(2,2)*DJ(3,1)) /DETJ
      DO 40 I=1,6
C
C FORM MATRIX  B(6X72), WHERE (B) = (BA)
C
      DJI(3,3) = (DJ(1,1)*DJ(2,2) - DJ(1,2)*DJ(2,1)) /DETJ
      DJI(3,2) = (DJ(3,1)*DJ(1,2) - DJ(3,2)*DJ(1,1)) /DETJ
      DO 40 L=1,72
   40 BA(I,L) = 0.D0
      DO 50 N=1,70,3
      L = (N - 1)/3 + 1
      BA(1,N) = (DJI(1,1)*C(1,L)+DJI(1,2)*C(2,L)+DJI(1,3)*C(3,L))/64.D0
      BA(4,N) = (DJI(2,1)*C(1,L)+DJI(2,2)*C(2,L)+DJI(2,3)*C(3,L))/64.D0
   50 BA(5,N) = (DJI(3,1)*C(1,L)+DJI(3,2)*C(2,L)+DJI(3,3)*C(3,L))/64.D0
      DO 60 N=2,71,3
      L = (N - 2)/3 + 1
      BA(2,N) = (DJI(2,1)*C(1,L)+DJI(2,2)*C(2,L)+DJI(2,3)*C(3,L))/64.D0
      BA(4,N) = (DJI(1,1)*C(1,L)+DJI(1,2)*C(2,L)+DJI(1,3)*C(3,L))/64.D0
   60 BA(6,N) = (DJI(3,1)*C(1,L)+DJI(3,2)*C(2,L)+DJI(3,3)*C(3,L))/64.D0
      DO 70 N=3,72,3
```

```
      L = (N - 3)/3 + 1
      BA(3,N) = (DJI(3,1)*C(1,L)+DJI(3,2)*C(2,L)+DJI(3,3)*C(3,L))/64.D0
      BA(5,N) = (DJI(1,1)*C(1,L)+DJI(1,2)*C(2,L)+DJI(1,3)*C(3,L))/64.D0
   70 BA(6,N) = (DJI(2,1)*C(1,L)+DJI(2,2)*C(2,L)+DJI(2,3)*C(3,L))/64.D0
C
C     FIND ELASTIC MATRIX AND FORM TRIPLE MATRIX PRODUCT
C
      IF(ILNP .EQ. 1 .OR.IX(INEL,27) .EQ. 2 .OR. IX(INEL,27) .EQ. 4)
     1     CALL ELASTR
      DO 80 N=1,6
      DO 80 L=1,72
      BDB(N,L) = 0.D0
      DO 80 NN=1,6
   80 BDB(N,L) = BDB(N,L) + D(N,NN)*BA(NN,L)
      DO 90 I=1,6
      DO 90 J=1,72
   90 SIG(I)=SIG(I) + BDB(I,J)*UEL(J)
C
C     INCLUDE THERMAL AFFECTS
C
      TMP = TMPND(IX(INEL,ILNP) ) - AMBTMP
      DO 92 I=1,6
      DO 92 J=1,6
   92 SIG(I) = SIG(I) - D(I,J)*ALFTMP(J) * TMP
      IF(ISTRS .EQ. 1 .OR. ISTRS .EQ. 3)  GO TO 96
C     PRINT STRESSES IN RECTANGULAR COORDINATES
      IF (LINE.LT.48) GO TO 94
      CALL TITLE
      WRITE (LIST,2000)
   94 CONTINUE
      IF( ILNP .EQ. 1 )  GO TO 95
      WRITE (LIST,2001) IX(INEL,ILNP), (SIG(I),I=1,6)
      IF(ISTRS .EQ. 0) GO TO 100
      IF(ISTRS .EQ. 4)  GO TO 96
   95 WRITE(LIST,2002)INEL,IX(INEL,ILNP),(SIG(I),I=1,6)
      IF(ISTRS .EQ. 0) GO TO 100
C
C     TRANSFORM STRESSES TO CYLINDRICAL COORDINATES
C
   96 SIGXY(1,1) = SIG(1)
      SIGXY(1,2) = SIG(4)/2.D0
      SIGXY(1,3) = SIG(5)/2.D0
      SIGXY(2,1) = SIG(4)/2.D0
      SIGXY(2,2) = SIG(2)
      SIGXY(2,3) = SIG(6)/2.D0
      SIGXY(3,1) = SIG(5)/2.D0
      SIGXY(3,2) = SIG(6)/2.D0
      SIGXY(3,3) = SIG(3)
      THEDA = 3.14159265358979D0 * YT(IX(INEL,ILNP)) / 180.D0
  540 T(1,1) = DCOS(THEDA)
      T(1,2) = DSIN(THEDA)
```

```
      T(1,3) = 0.D0
      T(2,1) = -T(1,2)
      T(2,2) = T(1,1)
      T(2,3) = 0.D0
      T(3,1) = 0.D0
      T(3,2) = 0.D0
      DO 537 II=1,3
      T(3,3) = 1.D0
      DO 537 JJ=1,3
  537 SIGRTH(II,JJ) = 0.D0
      DO 538 II=1,3
      DO 538 JJ=1,3
      DO 538 KK=1,3
      DO 538 LL=1,3
      IF(  DABS(T(II,KK))  .LT.  1.D-16  )    T(II,KK) = 0.D0
  538 SIGRTH(II,JJ) = SIGRTH(II,JJ) + T(II,KK)*T(JJ,LL)*SIGXY(KK,LL)
      SIG(1) = SIGRTH(1,1)
      SIG(2) = SIGRTH(2,2)
      SIG(3) = SIGRTH(3,3)
      SIG(4) = SIGRTH(1,2)*2.D0
      SIG(5) = SIGRTH(1,3)*2.D0
      SIG(6) = SIGRTH(2,3)*2.D0
C
C    PRINT STRESSES IN CYLINDRICAL COORDINATES
C
      IF(ISTRS .NE. 4)  GO TO 539
      DO 98 ISIG=1,6
   98 SIGSAV(ISIG,ILNP) = SIG(ISIG)
      IF(ILNP .LT. 24)  GO TO 100
      WRITE(LIST,2003)
      WRITE(LIST,2005)  INEL, IX(INEL,1), (SIGSAV(I,1),I=1,6)
      WRITE(LIST,2004)   (IX(INEL,J),(SIGSAV(I,J),I=1,6),J=2,24)
      LINE = LINE + 24
  539 IF (LINE.LT.48) GO TO 97
      CALL TITLE
      WRITE (LIST,2003)
   97 CONTINUE
      IF( ILNP .EQ. 1 )  GO TO 99
      WRITE (LIST,2004) IX(INEL,ILNP), (SIG(I),I=1,6)
      GO TO 100
   99 WRITE(LIST,2005)INEL,IX(INEL,ILNP),(SIG(I),I=1,6)
  100 CONTINUE
      LINE =LINE + 1
  200 CONTINUE
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELASTR
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                 *
C     *  SUBROUTINE ELASTR, IN CONJUNCTION WITH DMATST, CALCULATES THE  *
C     *  ELASTIC MATRIX FOR EACH ELEMENT                                *
C     *                                                                 *
C     *  THIS SUBROUTINE IS CALLED BY -                                 *
C     *        STRESS                                                   *
C     *                                                                 *
C     *  THIS SUBROUTINE CALLS -                                        *
C     *        DMATST                                                   *
C     *                                                                 *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, MTLND
      COMMON / GENLS /  NEL, NGLOF, NGNP, NMTL, INEL, ILNP, IGNP, IMTL
      COMMON /MATL /    E(9,9,10),D(6,6), FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9),AMBTMP    ,ETM(9), TMPEL(9,10), NTMP(9)
      COMMON  / NODAL /  X(1015), Y(1015), Z(1015), UX(1015),UY(1015),
     1   UZ(1015), TMPND(1015), U(3045), ALFTMP(6), XT(1015), YT(1015),
     2      ISTRS, IX(144,27), MTLND(1015)
      IF( IX(INEL,27) .EQ. 1  .OR.  IX(INEL,27) .EQ. 3)  GO TO 31
      IF( IX(INEL,27) .EQ. 2 )  GO TO 2
      NTMP1 = NTMP(IX(INEL,25))
      IF( NTMP1 .EQ. 1 )  GO TO 31
      IMTL = MTLND(IX(INEL,ILNP))
      IF(IMTL .EQ. 0 )  IMTL = IX(INEL,25)
      IGNT = IX(INEL,ILNP)
      IF(TMPND(IGNT) .LT. TMPEL(IMTL,1) )  GO TO 5
      IF( TMPND(IGNT) .GE. TMPEL(IMTL,NTMP(IMTL))  )  GO TO 6
      NTMPM1 = NTMP(IMTL) - 1
      DO 20 II=1,NTMPM1
      IF( TMPND(IGNT).GT. TMPEL(IMTL,II)  .AND.  TMPND(IGNT).LE.
     1          TMPEL(IMTL,II+1) ) GO TO 4
   20 CONTINUE
      WRITE(6,6001)
 6001 FORMAT( ' ERROR 1'   )
    5 DO 30 I=1,9
   30 ETM(I) = E(IMTL,I,1)
      GO TO 1
    6 DO 40 I=1,9
   40 ETM(I) = E(IMTL,I,NTMP(IMTL))
      GO TO 1
    4 DIFTP1 = TMPEL(IMTL,II+1) - TMPEL(IMTL,II)
      DIFTP2 = TMPND(IGNT) - TMPEL(IMTL,II)
      RATDIF = DIFTP2 / DIFTP1
      DO 50 I=1,9
   50 ETM(I) = E(IMTL,I,II) + RATDIF * (E(IMTL,I,II+1) - E(IMTL,I,II))
      GO TO 1
```

```
31 IMTL = IX(INEL,25)
   DO 60 I=1,9
60 ETM(I) = E(IMTL,I,1)
   GO TO 1
 2 IMTL = MTLNO(IX(INEL,ILNP))
   IF(IMTL .EQ. 0 ) IMTL = IX(INEL,25)
   DO 80 I=1,9
80 ETM(I) = E(IMTL,I,1)
 1 CALL DMATST
   RETURN
   END
```

```
      SUBROUTINE  DMATST
C
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    *                                                                   *
C    *   SUBROUTINE DMATST CALCULATES THE ELASTIC MATRIX AND PERFORMS A *
C    *   ROTATIONAL TRANSFORMATION ON THE ELASTIC MATRIX                 *
C    *                                                                   *
C    *   THIS SUBROUTINE IS CALLED BY -                                  *
C    *         ELASTR                                                    *
C    *                                                                   *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, MTLND
      COMMON / GENLS /  NEL, NCLDF, NGNP, NMTL, INEL, ILNP, IGNP, IMTL
      COMMON /MATL /    E(9,9,10),D(6,6), FIBORT(9), ALFA1(9),
     1   ALFA2(9), ALFA3(9),AMBTMP    ,ETM(9), TMPEL(9,10), NTMP(9)
      COMMON   / NODAL /  X(1015), Y(1015), Z(1015), UX(1015),UY(1015),
     1    UZ(1015), TMPND(1015), U(3045), ALFTMP(6), XT(1015), YT(1015),
     2       ISTRS, IX(144,27), MTLND(1015)
      DIMENSION         T(6,6),TD(6,6), TT(6,6), TMPCOF(6), DTMP(6,6)
      DO 10 I=1,6
      DO 10 J=1,6
      T(I,J) = 0.D0
      D(I,J) = 0D0
   10 TD(I,J) = 0.D0
      XNU21 = ETM(4) * ETM(2) / ETM(1)
      XNU31 = ETM(5) * ETM(3) / ETM(1)
      XNU32 = ETM(6) * ETM(3) / ETM(2)
      FACT=1.D0-ETM(4)*(XNU21+ETM(6)*XNU31)-ETM(5)*(XNU31+XNU32*XNU21)-
     1     ETM(6)*XNU32
      D(1,1)       = ETM(1) * (1.D0 - ETM(6) * XNU32) / FACT
      D(1,2)        = ETM(2) * (ETM(4) + ETM(5) * XNU32) / FACT
      D(1,3)        = ETM(3) * (ETM(5) + ETM(4) * ETM(6)) / FACT
      D(2,1)        = D(1,2)
      D(2,2)        = ETM(2) * (1.D0 - ETM(5) * XNU31) / FACT
      D(2,3)        = ETM(3) *(ETM(6) + ETM(5)* XNU21) / FACT
      D(3,1)        = D(1,3)
      D(3,2)        = D(2,3)
      D(3,3)        = ETM(3)* (1.D0 - ETM(4) * XNU21) / FACT
      D(4,4)        = ETM(7)
      D(5,5)        = ETM(8)
      D(6,6)        = ETM(9)
      ALFTMP(1) = ALFA1(IMTL)
      ALFTMP(2) = ALFA2(IMTL)
      ALFTMP(3) = ALFA3(IMTL)
      DO 60 I=4,6
   60 ALFTMP(I) = 0D0
      IF( DABS(FIBORT(IMTL)) .LT.  .5D-14 )  GO TO 50
      FIBOR = FIBORT(IMTL) * 3.14159265358979320D0 / 180.D0
      T(1,1) = DCOS(FIBOR )**2
```

```
      T(1,2) = DSIN(FIBOR )**2
      T(4,1) = DCOS(FIBOR )  *  DSIN(FIBOR )
      T(1,4) = -2.D0 *T(4,1)
      T(2,1) = T(1,2)
      T(2,2) = T(1,1)
      T(2,4) = -T(1,4)
      T(3,3) = 1.D0
      T(4,2) = -T(4,1)
      T(4,4) = T(1,1) - T(1,2)
      T(5,5) = DCOS(FIBOR )
      T(6,5) = DSIN(FIBOR )
      T(5,6) = -T(6,5)
      T(6,6) = T(5,5)
      DO 70 I=1,6
      TMPCOF(I) = 0D0
      DO 70 J=1,6
   70 TMPCOF(I) = TMPCOF(I) + T(I,J) * ALFTMP(J)
      DO 90 I=1,6
   90 ALFTMP(I) = TMPCOF(I)
      DO 20 I=1,6
      DO 20 J=1,6
      DO 20 K=1,6
   20 TD(I,J) = TD(I,J) + T(I,K)*D(K,J)
      DO 80 I=1,6
      DO 80 J=1,6
   80 TT(J,I) = T(I,J)
      DO 30 I=1,6
      DO 30 J=1,6
      DTMP(I,J) = 0D0
      DO 30 K=1,6
   30 DTMP(I,J) = DTMP(I,J) + TD(I,K) * TT(K,J)
      DO 40 I=1,6
      DO 40 J=1,6
   40 D(I,J)       = DTMP(I,J)
   50 CONTINUE
      RETURN
      END
```

APPENDIX B

Input/Output Units and Sample JCL

A. Introduction

One disk unit is required for job steps 2, 3 and 4. This unit is a direct access file with a minimum logical record length of 21024 bytes. This file is used to pass data from step 2 to step 3 and from step 3 to step 4. It is also used in the iterative loop in step 3.

A sequential disk or tape unit can be used to pass data from a mesh generator (step 1) to step 2. This sequential unit should be blocked for card images.

The sample JCL given in this Appendix is for the IBM 360/370 operating system.

JCL to run four job steps

```
JOB URIGIN FRUM LOCAL DEVICE=RD1   .   ,2CC.
//R1225TS1 JOB F0803,DANA,MSGLEVEL=1
/*MAIN   REGION=(200,292,230,248),TIME=(1,2,30,4),LINES=4
//STEP1 EXEC FORTGCG
//FORT.SYSIN DD *
/*
//GO.FT01F001 DD  DSN=&DATATEM,UNIT=SYSDA,
//     SPACE=(TRK,(5,1),RLSE),
//     DISP=(NEW,PASS,DELETE),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.SYSIN DD *
/*
//STEP2 EXEC FORTGCG
//FORT.SYSLIN DD DSN=&LOADSET,DISP=(NEW,PASS),
//  UNIT=SYSSQ,SPACE=(80,(200,150),RLSE),DCB=BLKSIZE=80
//FORT.SYSIN DD *
//GO.FT01F001 DD  DSN=&DATATEM,UNIT=SYSDA,
//     DISP=(OLD,DELETE)
//GO.FT03F001 DD  DSN=&KEMTREC,UNIT=SYSDA,
//      DISP=(NEW,PASS,DELETE),
//     SPACE=(26000,(15,9),RLSE),
//     DCB=(RECFM=FT,LRECL=26000,BLKSIZE=26000,BUFNO=1)
//GO.SYSIN DD *
/*
//STEP3 EXEC FORTGCG
//FORT.SYSIN DD *
/*
//GO.FT03F001 DD  DSN=&KEMTREC,UNIT=SYSDA,
//     DISP=(OLD,PASS,DELETE),
//     SPACE=(26000,(15,9),RLSE),
//     DCB=(RECFM=FT,LRECL=26000,BLKSIZE=26000,BUFNO=1)
//GO.SYSIN DD *
/*
//STEP4 EXEC FORTGCG
//FORT.SYSLIN DD DSN=&LOADSET,DISP=(NEW,PASS),
//  UNIT=SYSSQ,SPACE=(80,(200,150),RLSE),DCB=BLKSIZE=80
//FORT.SYSIN DD *
/*
//GO.FT03F001 DD  DSN=&KEMTREC,UNIT=SYSDA,
//     SPACE=(26000,(15,9),RLSE),
//      DISP=(OLD,DELETE,DELETE),
//     DCB=(RECFM=FT,LRECL=26000,BLKSIZE=26000,BUFNO=1)
//GO.SYSIN DD *
/*
//
```

JCL to run steps 1 and 2

```
JOB ORIGIN FROM LOCAL DEVICE=RD1      ,20C.
//P1225T17 JOB F0803,DANA,MSGLEVEL=1
/*MAIN      REGION=(,200,292),TIME=(1,1,5),LINES=4
//  EXEC PGM=IEFBR14
//DD1 DD DISP=(OLD,DELETE),DSN=JOND.AF0803,UNIT=SYSDA,VOL=SER=USERPK
/*
//STEP1 EXEC FORTGCG
//FORT.SYSIN DD *
//GO.FT01F001 DD  DSN=&DATATEM,UNIT=SYSDA,
//     SPACE=(TRK,(5,1),RLSE),
//     DISP=(NEW,PASS,DELETE),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.SYSIN DD *
/*
//STEP2 EXEC FORTGCG
//FORT.SYSLIN DD DSN=&LOADSET,DISP=(NEW,PASS),
//  UNIT=SYSSQ,SPACE=(80,(200,150),RLSE),DCB=BLKSIZE=80
//FORT.SYSIN DD *
//GO.FT01F001 DD  DSN=&DATATEM,UNIT=SYSDA,
//     DISP=(OLD,DELETE)
//GO.FT03F001 DD  DSN=JOND.AF0803,UNIT=SYSDA,VOL=SER=USERPK,
//     DISP=(NEW,PASS,DELETE),
//     SPACE=(26000,(15,3),RLSE),
//     DCB=(RECFM=FT,LRECL=26000,BLKSIZE=26000,LUFNO=1)
//GO.SYSIN DD *
/*
```

JCL to run step 3

```
JOB ORIGIN FROM LOCAL DEVICE=RD1      ,20C.
//B1225T13 JOB FC803,DANA,MSGLEVEL=1
/*MAIN   REGION=230,TIME=30,LINES=4
//STEP3 EXEC FORTGCG
//FORT.SYSIN DD *
/*
//GO.FT03F001 DD  DSN=JOND.AF0803,UNIT=SYSDA,VOL=SER=USERPK,
//     DISP=(OLD,KEEP,KEEP),
//     SPACE=(26000,(15,3),RLSE),
//     DCB=(RECFM=FT,LRECL=26000,BLKSIZE=26000,BUFNO=1)
//GO.SYSIN DD *
/*
//
```

JCL to run step 4

```
JOB ORIGIN FROM LOCAL DEVICE=RD1      ,20C.
//B1225T16 JOB FC803,DANA,MSGLEVEL=1
//STEP4 EXEC FORTGCG
//FORT.SYSLIN DD DSN=&LOADSET,DISP=(NEW,PASS),
//  UNIT=SYSSQ,SPACE=(80,(200,150),RLSE),DCB=BLKSIZE=80
//FORT.SYSIN DD *
/*
//GO.FT03F001 DD  DSN=JOND.AF0803,UNIT=SYSDA,VOL=SER=USERPK,
//     DISP=(OLD,KEEP,KEEP),
//     SPACE=(26000,(15,3),RLSE),
//     SPACE=(26000,(15,3),RLSE),
//     DCB=(RECFM=FT,LRECL=26000,BLKSIZE=26000,BUFNO=1)
//GO.SYSIN DD *
/*
//
```

APPENDIX C

Rectangular Plate Mesh Generator

A. Introduction

This mesh generator will yield element, nodal, material, and temperature distribution data necessary to idealize a rectangular solid laminate subjected to plate bending loads or axial extension in the x, y or z-directions. Force and displacement boundary conditions for each node are generated by specifying the boundary condition codes and values at a point, along a line or on a plane. From one to six elements can be specified in the x- or y-directions and from one to ten elements can be specified in the z-direction. Figure C-1 shows a typical mesh.

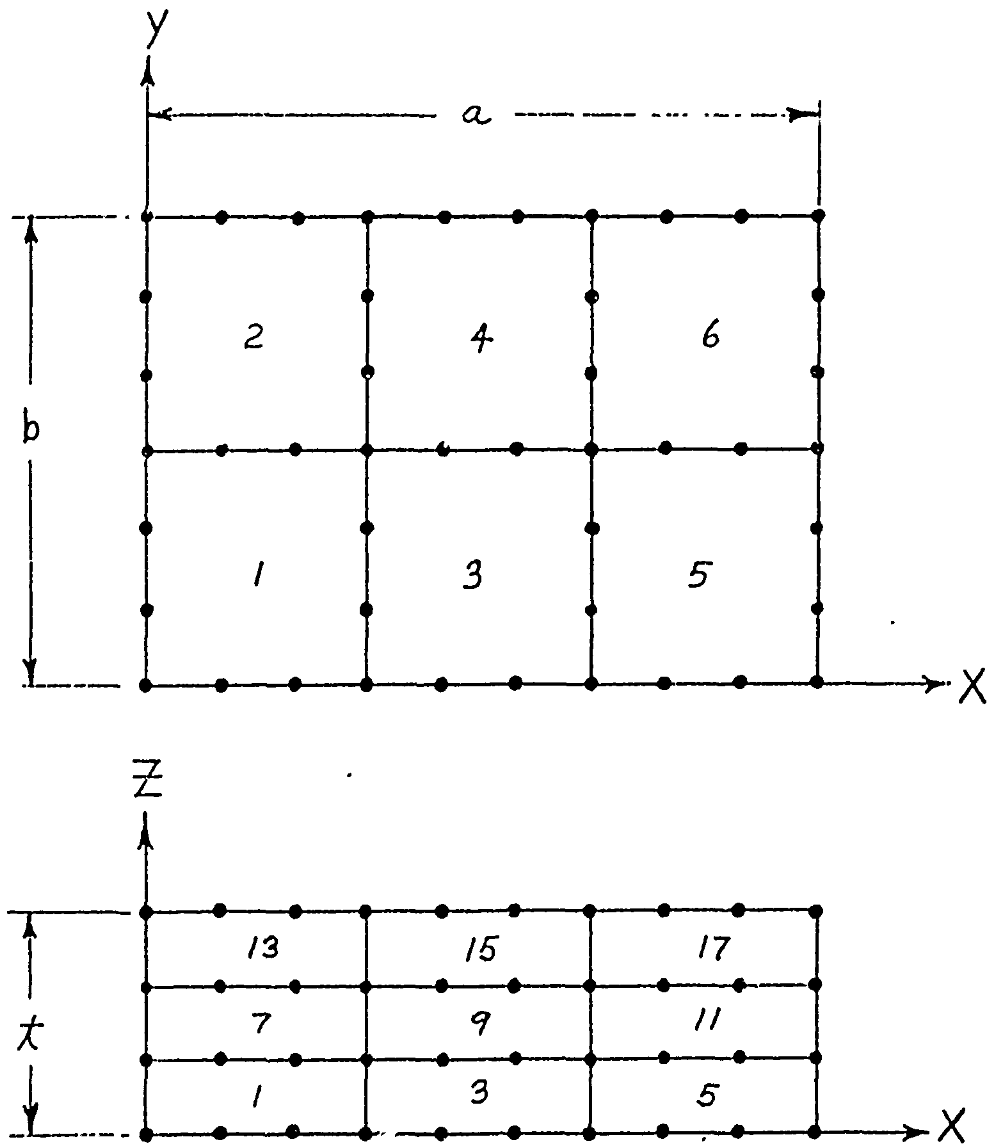

FIGURE C-1: Mesh for Rectangular Plate

B. Input Data

1. Heading Card (10A8)

   Columns 1-80 information to be printed with output

2. Output unit card (16I5)

   Columns 1-5 unit number (specifies the unit for passing data to the next job step, e.g. unit seven if data is to be punched on cards)

3. Control parameters cards two cards

   First card (2I5, F10.2)

   Columns 1-5 type of thermal-elastic problem (called classes)

   '1' - elastic only, constant material properties within an element

   '2' - elastic only, material properties can vary within an element

   '3' - thermal elastic, material properties cannot vary with temperature within an element

   '4' - thermal elastic, material properties can vary with temperature within an element

   (Class 1 or 2 elements cannot be mixed with Class 3 or 4 elements. Classes 1 and 2 can be mixed and classes 3 and 4 can be mixed.)

   6-10 type of temperature distribution

   '0' - constant temperature

   '1' - one dimensional variation, x-direction

   '2' - one dimensional variation, y-direction

   '3' - one dimensional variation, z-direction

   '4' - two dimensional variation, x-y plane

   '5' - two dimensional variation, x-z plane

   '6' - two dimensional variation, y-z plane

   '7' - three dimensional variation (not used)

   11-20 initial temperature

Second card (I10, 2F10.5) blank card if class 1 or 2.

Columns 1-10 maximum number of iterations for finding temperature distribution

11-20 initial guess for temperature distribution

21-30 accuracy of temperature distribution

4. Plate and mesh dimensions card (3(I5, F10.0)) one card

Columns 1-5 number of elements in the x-direction ('1' to '6')

6-15 x-dimension of plate, a

16-20 number of elements in the y-direction ('1' to '6')

21-30 y-dimension of plate, b

31-35 number of elements in the z-direction ('1' to '6')

36-45 z-dimension of plate, t

5. Material change data cards (16I5)

Columns 1-5 number of materials

5-10 number of material changes

11-15 material number

16-20 element number at which the material changed (Use as many sets of material number and element number as required to describe at which element a material is changed. The elements are numbered, on the plate, first in the y-direction, then the x-direction and then in the z-direction.)

6. Material data cards two cards

First card (2I5, 7F10.0) one for each material

Columns 1-5 material number (in sequential order)

6-10 number of material cards ('1' for class 1, 2 or 3)

11-20 fiber orientation in degrees

21-30 thermal expansion coefficient, α_{11}

31-40 thermal expansion coefficient, α_{22}

41-50 thermal expansion coefficient, α_{33}

Subsequent cards (F5.0, 3F10.0, 3F5.2, 3F10.0) (One card for problem class 1, 2 or 3. And for problem class 4, one

card for each temperature for which material properties are specified.)

Columns 1-5 temperature for material properties (can be left blank for class 1 and 2 problems)

6-15 modulus of elasticity, E_{11}, KSI

16-25 modulus of elasticity, E_{22}, KSI

26-35 modulus of elasticity, E_{33}, KSI

36-40 Poisson's ratio, ν_{12}

41-45 Poisson's ratio, ν_{13}

46-50 Poisson's ratio, ν_{23}

51-60 shear modulus, G_{12} KSI

61-70 shear modulus, G_{13} KSI

71-80 shear modulus, G_{23} KSI

7. Element change data cards (16I5)

Columns 1-5 number of unique elements

6-10 number of element changes

11-15 element type number

16-20 element number at which the element type changes (Use as many sets of element type and element number as required to describe at which element number an element type is changed.)

8. z-direction load coefficients (8F10.0) one card (leave blank if no loads in the z-direction)

The load in the z-direction is evaluated from the following expression.

$$P = C_1 + C_2 x + C_3 y + C_4 x^2 + C_5 y^2 + C_6 xy + C_7\left(\sin \frac{\pi x}{A}\right)\left(\sin \frac{\pi y}{B}\right) + C_8\left(\sin \frac{\pi x}{2A}\right)\left(\sin \frac{\pi y}{2B}\right)$$

Where:

Columns 1-10 C_1, constant coefficient

11-20 C_2, coefficient for linear variation in the x-direction

21-30 C_3, coefficient for linear variation in the y-direction

31-40 C_4, coefficient for quadratic variation in the x-direction

41-50 C_5, coefficient for quadratic variation in the y-direction

51-60 C_6, coefficient for product variation in x and y directions

61-70 C_7, coefficient for full sine function in x and y directions

71-80 C_8, coefficient for half sine function in x and y directions

9. Temperature boundary conditions (8F10.0) one card

(Leave blank for class 1 or 2)

Columns 1-10 temperature at corner 1 (x, y, z) = (0, 0, 0)

11-20 temperature at corner 2 (x, y, z) = (a, 0, 0)

21-30 temperature at corner 3 (x, y, z) = (0, b, 0)

31-40 temperature at corner 4 (x, y, z) = (a, b, 0)

41-50 temperature at corner 5 (x, y, z) = (0, 0, t)

51-60 temperature at corner 6 (x, y, z) = (a, 0, t)

61-70 temperature at corner 7 (x, y, z) = (0, b, t)

71-80 temperature at corner 8 (x, y, z) = (a, b, t)

10. Material properties at nodes

First Card (4I5, 3F10.0) blank card for class 1, 3 or 4

Columns 1-5 number of material property cards

Second Card (4I5, 3F10.0) no cards for class 1, 3 or 4

Columns 1-5 material number

6-10 index coordinate, x-direction

11-15 index coordinate, y-direction

16-20 index coordinate, z-direction

11. Force/displacement boundary conditions (4I5, 3F10.0)

Columns 1-5 boundary condition code

6-10 index coordinate, x-direction
11-15 index coordinate, y-direction
16-20 index coordinate, z-direction
21-30 magnitude of x-direction boundary condition
31-40 magnitude of y-direction boundary condition
41-50 magnitude of z-direction boundary condition

```
C       MAIN PROGRAM STEP 1A
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                               *
C     *  THIS MESH GENERATOR IDEALIZES A RECTANGULAR LAMINATED PLATE  *
C     *                                                               *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, ICODE, IDPIX1, MTLND
      COMMON /GENMAT/ X(1066),Y(1066),Z(1066),UX(1066),UY(1066),UZ(1066)
     1 ,  TMPND(1066), BCTMP(8),
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1066),IDPIX1(19,19,11),MTLND(1066)
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1    NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2    , ITYTD, NELX31, NELY31, NELZ1
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
 1000 FORMAT( 16I5)
 1001 FORMAT(2I5,  F10.2, 3F10.0)
 1002 FORMAT(F5.0, 3F10.0, 3F5.2, 3F10.0)
 1003 FORMAT( I4, I4, I2, 6F10.0, F10.2)
 1004 FORMAT( 10A8)
 1005 FORMAT( 4I5, F10.2)
 1011 FORMAT(I10, 2F10.5)
      ICRD = 5
      IPAGE = 1
      IWRT = 6
      READ(5,1004)   HED
      READ(5,1000)   NTUT
      READ(5,1001)   ICLASS, ITYTD, AMBTMP
      READ(5,1011)                          LMTMP, TMPINT, EPSTMP
C
C     GENERATE MESH AND BOUNDARY CONDITIONS
C
      CALL MSHGEN
      CALL BCGEN
      DO 20 I=1,NGNP
   20 TMPND(I) = AMBTMP
      IF ( ICLASS .LE. 2 )   GO TO 1
      CALL TMPDST
    1 CONTINUE
      CALL DISPLY
      CALL MODF
C
C     WRITE MESH DATA ON UNIT NTUT
C
      WRITE(NTUT,1004) HED
      WRITE(NTUT,1005)   NGNP, NMTL, NEL, NTYEL, AMBTMP
      DO 10 IMTL=1,NMTL
      WRITE(NTUT,1001)   IMTL, NTMP(IMTL), FIBORT(IMTL), ALFA1(IMTL),
```

```
     1       ALFA2(IMTL), ALFA3(IMTL)
      NTMP1 = NTMP(IMTL)
      DO 10 ITMP=1,NTMP1
   10 WRITE(NTUT,1002)   TMPEL(IMTL,ITMP), (E(IMTL,J,ITMP),J=1,9)
      DO 30 INEL=1,NEL
   30 WRITE(NTUT,1000)  INEL,  (IX(INEL,J),J=1,27)
      DO 40    M=1,NGNP
   40 WRITE(NTUT,1003)  M, MTLND(M), ICODE(M), X(M), Y(M), Z(M),
     1    UX(M), UY(M), UZ(M), TMPND(M)
      IF( NTUT .NE. 7)   REWIND  NTUT
      STOP
      END
```

```
      SUBROUTINE   MSHGEN
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   *  SUBROUTINE MSHGEN GENERATES THE MESH AND NUMBERS NODES AND     *
C   *  ELEMENTS                                                       *
C   *                                                                 *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, ICODE, IDPIX1, MTLND
      COMMON /GENMAT/ X(1066),Y(1066),Z(1066),UX(1066),UY(1066),UZ(1066)
     1 ,  TMPND(1066), BCTMP(8),
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1066),IDPIX1(19,19,11),MTLND(1066)
      COMMON  / GENL1 /   TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1     NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2    , ITYTD, NELX31, NELY31, NELZ1
      COMMON / CORD / RUNX, RUNY, RUNZ, IRUNX, IRUNY, IRUNZ, IST
      COMMON /DATDIS/ XCRD(19), YCRD(19), ZCRD(11)
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
      DIMENSION   IMATL(10), IXMLCH(10), ITYEL(10), IXELCH(10)
      DIMENSION  INEWX(16), IRECX(8), IPSSX(8)
      DIMENSION  INEWZ1(12), IRECZ(12),            IPSSY(4), IRECY(4),
     1     INEWZ3(8), IPSSXZ(4), IRECXZ(4)
      DIMENSION     ISG1(3), ISG2(3), ISG3(3), ISG4(3)
      DIMENSION     LNCRD(12), ISTCDX(12), ISTCDY(12)
      DATA INEWX/ 2,3,4,6,7,8,11,12,15,16,18,19,20,22,23,24 /
      DATA IRECX/ 1,5,9,10,13,14,17,21 /
      DATA IPSSX/ 4,8,11,12,15,16,20,24 /
      DATA   INEWZ1 / 5,6,7, 8, 13, 14, 15, 16, 21, 22, 23, 24 /
      DATA   IRECZ  / 1, 2, 3, 4, 9, 10, 11, 12, 17, 18, 19, 20 /
      DATA     IPSSY      / 21, 22, 23, 24 /
      DATA   IRECY / 5, 6, 7, 8  /
      DATA   INEWZ3 /  6, 7, 8, 15, 16, 22, 23, 24 /
      DATA   IPSSXZ / 8, 15, 16, 24 /
      DATA   IRECXZ / 5, 13, 14, 21 /
      DATA   ISG1, ISG2, ISG3, ISG4 /8,1,7,7,1,8,4,1,5,3,2,2/
      DATA   LNCRD  /5, 6, 7, 8, 13, 14, 15, 16, 21, 22, 23, 24 /
      DATA   ISTCDX  /0, 1, 1, 1, -3, 0, 3, 0, -3, 1, 1, 1 /
      DATA   ISTCDY  / 0, 0, 0, 0, 1, 1, -1, 1, 1, 0, 0, 0 /
  100 FORMAT( 16I5)
  101 FORMAT( 3(I5 , F10.0))
  204 FORMAT('OPLATE DIMENSIONS', T40, 'X =', F9.3, '     Y =', F9.3,
     1       '     Z =',  F9.3  )
  205 FORMAT(' NUMBER OF ELEMENT', T40, 'XN =', I3, '      YN =', I3,
     1       '       ZN =', I3)
  206 FORMAT('OMATERIAL TYPE AND MATERIAL CHANGES'  /
     1    T10,  'MATERIAL TYPE          CHANGE AT ELEMENT'  )
  306 FORMAT(18X, I2, 22X, I4)
  307 FORMAT('OELEMENT TYPE AND ELEMENT CHANGES'  /
```

```
    1   T10,  'ELEMENT TYPE           CHANGE AT ELEMENT'  )
1001 FORMAT(2I5, 7F10.0)
1002 FORMAT(F5.0, 3F10.0, 3F5.0, 3F10.0)
     READ(ICRD,101)  NELX, AX, NELY, BY, NELZ, CZ
     READ(ICRD,100)   NMTL, NMLCH, (IMATL(J),IXMLCH(J),J=1,NMLCH)
     DO 71 IMTL=1,NMTL
     READ(ICRD ,1001)  MTLN, NTMP(IMTL), FIBORT(IMTL), ALFA1(IMTL),
    1        ALFA2(IMTL), ALFA3(IMTL)
     NTMP1 = NTMP(IMTL)
     DO 71 ITMP=1,NTMP1
  71 READ(ICRD ,1002)   TMPEL(IMTL,ITMP), (E(IMTL,J,ITMP),J=1,9)
     READ(ICRD,100)  NTYEL, NELCH, (ITYEL(J),IXELCH(J),J=1,NELCH)
     CALL TITLE
     WRITE(IWRT,204)       AX, BY, CZ
     WRITE(IWRT,205)       NELX, NELY, NELZ
     WRITE(IWRT,206)
     WRITE(IWRT,306) (IMATL(J), IXMLCH(J), J=1,NMLCH)
     WRITE(IWRT,307)
     WRITE(IWRT,306)  (ITYEL(J), IXELCH(J),J=1,NELCH)
     INEL = 0
     DO 1   IELY=1,NELY
     INEL = INEL+1
     IELY1= IELY-1
     DO 1   J=1,24
     IGNP = J+16*IELY1
   1 IX(INEL,J) = IGNP
     IF(NELX .EQ. 1 )  GO TO 6
     DO 5   IELX=2,NELX
     IGNPL = IGNP
     DO 3   IELY=1,NELY
     INEL = INEL + 1
     IMNY = INEL-NELY
     DO 2   J=1,8
   2 IX(INEL,IRECX(J)) = IX(IMNY,IPSSX(J))
   3 CONTINUE
     INEL = INEL- NELY
     DO 5   IELY=1,NELY
     INEL = INEL+1
     IELY1= IELY-1
     DO 4   J=1,16
     IGNP = IGNPL+J+10*IELY1
   4 IX(INEL,INEWX(J)) = IGNP
   5 CONTINUE
   6 CONTINUE
     IF(NELZ .EQ. 1 )  GO TO 59
     DO 22  IELZ = 2,NELZ
     INEL = INEL+1
     IMNZ = INEL-NELX*NELY
     DO   7   J=1,12
   7 IX(INEL,IRECZ(J)) = IX(IMNZ,INEWZ1(J))
     DO   8   J=1,12
```

```
      IGNP = IGNP+1                                                     MSP 1030
    8 IX(INEL,INEWZ1(J)) = IGNP                                         MSP 1040
      IF(NELY .EQ. 1)  GO TO 66                                         MSP 1050
      DO 12  IELY =2,NELY                                               MSP 1060
      INEL = INEL+1                                                     MSP 1070
      IMNZ = INEL-NELX*NELY                                             MSP 1080
      DO  9  J=1,12                                                     MSP 1090
    9 IX(INEL,IRECZ(J)) = IX(IMNZ,INEWZ1(J))                            MSP 1100
      DO  10 J=1,4                                                      MSP 1110
   10 IX(INEL,IRECY(J)) = IX(INEL-1,IPSSY (J))                          MSP 1120
      DO 11  J=5,12                                                     MSP 1130
      IGNP = IGNP+1                                                     MSP 1140
   11 IX(INEL,INEWZ1(J)) = IGNP                                         MSP 1150
   12 CONTINUE                                                          MSP 1160
   66 CONTINUE                                                          MSP 1170
      IF(NELX .EQ. 1)  GO TO 22                                         MSP 1180
      DO 21  IELX =2,NELX                                               MSP 1190
      INEL = INEL+1                                                     MSP 1200
      IMNZ = INEL-NELX*NELY                                             MSP 1210
      DO 13  J=1,12                                                     MSP 1220
   13 IX(INEL,IRECZ(J)) = IX(IMNZ,INEWZ1(J))                            MSP 1230
      IMNX = INEL-NELY                                                  MSP 1240
      DO 14  J=1,4                                                      MSP 1250
   14 IX(INEL,IRECXZ(J)) = IX(IMNX,IPSSXZ(J))                           MSP 1260
      DO 15  J=1,8                                                      MSP 1270
      IGNP = IGNP+1                                                     MSP 1280
   15 IX(INEL,INEWZ3(J)) = IGNP                                         MSP 1290
      IF(NELY .EQ. 1)  GO TO 21                                         MSP 1300
      DO 20  IELY = 2,NELY                                              MSP 1310
      INEL = INEL+1                                                     MSP 1320
      IMNZ = INEL-NELX*NELY                                             MSP 1330
      IMNX = INEL-NELY                                                  MSP 1340
      DO 16  J=1,12                                                     MSP 1350
   16 IX(INEL,IRECZ(J)) = IX(IMNZ,INEWZ1(J))                            MSP 1360
      DO 17  J=1,4                                                      MSP 1370
   17 IX(INEL,IRECXZ(J)) = IX(IMNX,IPSSXZ(J))                           MSP 1380
      DO 18  J=2,4                                                      MSP 1390
   18 IX(INEL,IRECY(J)) = IX(INEL-1,IPSSY (J))                          MSP 1400
      DO 19  J=4,8                                                      MSP 1410
      IGNP = IGNP+1                                                     MSP 1420
   19 IX(INEL,INEWZ3(J)) = IGNP                                         MSP 1430
   20 CONTINUE                                                          MSP 1440
   21 CONTINUE                                                          MSP 1450
   22 CONTINUE                                                          MSP 1460
   59 NEL = INEL                                                        MSP 1470
      IXELCH(NELCH+1) = 0                                               MSP 1480
      IXMLCH(NMLCH+1) = 0                                               MSP 1490
      I = 0                                                             MSP 1500
      J = 0                                                             MSP 1510
      DO 23 INEL=1,NEL                                                  MSP 1520
      IF(INEL .EQ. IXMLCH(I+1))   I=I+1                                 MSP 1530
```

```
      IF(INEL .EQ. IXELCH(J+1))   J=J+1
      IX(INEL,25) = IMATL(I)
      IX(INEL,26) = ITYEL(J)
      IX(INEL,27) = ICLASS
   23 CONTINUE
      ELLX = AX/NELX
      ELLY = BY/NELY
      ELLZ = CZ/NELZ
      THELLX = ELLX/3.
      THELLY = ELLY/3.
      IRUNX = 1
      NELX31 = NELX*3+1
      NELY31 = NELY*3+1
      NELZ1 = NELZ+1
      IRUNY = 1
      IRUNZ = 1
      RUNX = 0D0
      RUNY = 0D0
      RUNZ = ELLZ
      IGNP = 1
      IST = 4
      DO 24 I=1,NELX31
      DO 24 J=1,NELY31
      DO 24 K=1,NELZ1
   24 IDPIX1(I,J,K) = 0
      XCRD(1) = 0.0
      YCRD(1) = 0.0
      ZCRD(1) = 0.0
      DO 25 I=2,NELX31
   25 XCRD(I) = XCRD(I-1) + THELLX
      DO 26 I=2,NELY31
   26 YCRD(I) = YCRD(I-1) + THELLY
      DO 27 I=1,NELZ
   27 ZCRD(I+1) = ZCRD(I) + ELLZ
      DO 28 IELY=1,NELY
      DO 28 J=1,3
      CALL COORD(IGNP)
      IGNP = IGNP + ISG1(J)
      IRUNY = IRUNY+1
   28 RUNY = RUNY + THELLY
      CALL COORD(IGNP)
      DO 30 I=1,2
      IRUNX = IRUNX+1
      RUNX = RUNX + THELLX
      IGNP = 1+I
      IRUNY = 1
      RUNY = 0D0
      DO 29 IELY=1,NELY
      CALL COORD(IGNP)
      IGNP = IGNP + 16
      IRUNY = IRUNY+3
```

```
29 RUNY = RUNY +    ELLY                                        MSP 2050
   CALL COORD(IGNP)                                             MSP 2060
30 CONTINUE                                                     MSP 2070
   IRUNX = IRUNX+1                                              MSP 2080
   RUNX = RUNX + THELLX                                         MSP 2090
   IGNP = 4                                                     MSP 2100
   IRUNY = 1                                                    MSP 2110
   RUNY = 0D0                                                   MSP 2120
   DO 31 IELY=1,NELY                                            MSP 2130
   DO 31 J=1,3                                                  MSP 2140
   CALL COORD(IGNP)                                             MSP 2150
   IGNP = IGNP + ISG2(J)                                        MSP 2160
   IRUNY = IRUNY+1                                              MSP 2170
31 RUNY = RUNY + THELLY                                         MSP 2180
   CALL COORD(IGNP)                                             MSP 2190
   IGNP = IGNP + 1                                              MSP 2200
   IST = 3                                                      MSP 2210
   IF(NELX .EQ. 1 )  GO TO 61                                   MSP 2220
   DO 35 IELX=2,NELX                                            MSP 2230
   IGNPL = IGNP+ 3                                              MSP 2240
   DO 33 I=1,2                                                  MSP 2250
   IRUNX = IRUNX+1                                              MSP 2260
   RUNX = RUNX + THELLX                                         MSP 2270
   IGNP = IGNPL + I                                             MSP 2280
   IRUNY = 1                                                    MSP 2290
   RUNY = 0D0                                                   MSP 2300
   DO 32 IELY=1,NELY                                            MSP 2310
   CALL COORD(IGNP)                                             MSP 2320
   IGNP = IGNP + 10                                             MSP 2330
   IRUNY = IRUNY+3                                              MSP 2340
32 RUNY = RUNY +    ELLY                                        MSP 2350
   CALL COORD(IGNP)                                             MSP 2360
33 CONTINUE                                                     MSP 2370
   IRUNX = IRUNX+1                                              MSP 2380
   RUNX = RUNX + THELLX                                         MSP 2390
   IGNP = IGNPL + 3                                             MSP 2400
   IRUNY = 1                                                    MSP 2410
   RUNY = 0D0                                                   MSP 2420
   DO 34 IELY=1,NELY                                            MSP 2430
   DO 34 J=1,3                                                  MSP 2440
   IDPIX1(IRUNX,IRUNY,1) = IGNP                                 MSP 2450
   IDPIX1(IRUNX,IRUNY,2)=IGNP+ISG4(J)                           MSP 2460
   X(IGNP   ) = RUNX.                                           MSP 2470
   X(IGNP+ISG4(J)) = RUNX                                       MSP 2480
   Y(IGNP   ) = RUNY                                            MSP 2490
   Y(IGNP+ISG4(J)) = RUNY                                       MSP 2500
   Z(IGNP   ) = 0D0                                             MSP 2510
   Z(IGNP+ISG4(J)) = RUNZ                                       MSP 2520
   IGNP = IGNP + ISG3(J)                                        MSP 2530
   IRUNY = IRUNY+1                                              MSP 2540
34 RUNY = RUNY + THELLY                                         MSP 2550
```

```
 35 CALL COORD(IGNP)
 61 CONTINUE
    INEL = NELX*NELY
    IF(NELZ .LT. 2)  GO TO 37
    DO 36  IELZ=2,NELZ
    IRUNX = 1
    IRUNZ = IELZ+1
    DO 36  IELX=1,NELX
    IRUNX = IRUNX + 3
    IRUNY = 1
    DO 36  IELY=1,NELY
    IRUNX = IRUNX-3
    INEL = INEL + 1
    IMNZ = INEL-NELY*NELX
    DO 36  J=1,12
    L = LNCRD(J)
    X(IX(INEL,L)) = X(IX(IMNZ,L))
    Y(IX(INEL,L)) = Y(IX(IMNZ,L))
    Z(IX(INEL,L)) = IELZ*ELLZ
    IRUNX = IRUNX + ISTCDX(J)
    IRUNY = IRUNY + ISTCDY(J)
 36 IDPIX1(IRUNX,IRUNY,IRUNZ) = IX(INEL,L)
 37 NELZ1 = NELZ+1
    NGNP =(IGNP + 3 ) * NELZ1/2
    DO 38 I=1,NGNP
 38 MTLND(I) = 0
    NGLDF = 3*NGNP
    RETURN
    END
```

```
      SUBROUTINE   COORD(IGNP)
C
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    *                                                                *
C    *  SUBROUTINE COORD CALCULATES THE X,Y&Z-COORDINATES FOR EACH NODE*
C    *                                                                *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, ICODE, IDPIX1, MTLND
      COMMON /GENMAT/ X(1066),Y(1066),Z(1066),UX(1066),UY(1066),UZ(1066)
     1 ,  TMPND(1066), BCTMP(8),
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1066),IDPIX1(19,19,11),MTLND(1066)
      COMMON / CORD / RUNX, RUNY, RUNZ, IRUNX, IRUNY, IRUNZ, IST
      IDPIX1(IRUNX,IRUNY,1)=IGNP
      IDPIX1(IRUNX,IRUNY,2)=IGNP+IST
      X(IGNP   ) = RUNX
      X(IGNP+IST)=RUNX
      Y(IGNP   ) = RUNY
      Y(IGNP+IST)=RUNY
      Z(IGNP   ) = 0D0
      Z(IGNP+IST)=RUNZ
      RETURN
      END
```

```
      SUBROUTINE  BCGEN
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE BCGEN GENERATES FORCE AND DISPLACEMENT BOUNDARY       *
C   *  CONDITION CODES                                                  *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, ICODE, IDPIX1, MTLNO
      COMMON /GENMAT/ X(1066),Y(1066),Z(1066),UX(1066),UY(1066),UZ(1066)
     1 ,  TMPND(1066), BCTMP(8),
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1066),IDPIX1(19,19,11),MTLNO(1066)
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1    NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2    , ITYTD, NELX31, NELY31, NELZ1
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
      DIMENSION  ZLCOEF(8)
      DIMENSION  LDRLVT(12,2), PBAR(24), VLDMAT(78), P(1066), RELVT(78)
      DATA LDRLVT / 1,2,3,4,17,18,19,20,9,10,11,12,
     1              5,6,7,8,21,22,23,24,13,14,15,16 /
      DATA  RELVT  /  4126., -4824., -2421., 2864., 2864., -3546.,
     1    -1494., 2116., -4824., -2421., -3546., -1494.,
     2    12960., -1620., -2421., -3546., 6480., -810., -1494., 9072.,
     3    2268., 2268., 567.,
     4    12960., -4824., -1494., -810., 6480., -3546., 2268.,567.,
     5    9072., 2268.,
     6    4126., 2116., -1494., -3546., 2864., -3546., -1494., -4824.,
     7    -2421.,
     8    4126., -4824., -2421., 2864., -2421., -4824., -1494., -3546.,
     9    12960., -1620., -2421., 2268., 9072., 567., 2268.,
     A    12960., -4824., 567.,2268., 2268., 9072.,
     B    4126., -1494., -3546., -2421., -4824.,
     C    12960., -1620., 6480., -810.,     12960., -810., 6480.,
     D    12960., -1620.,              12960. /
  102 FORMAT(8F10.0)
  103 FORMAT(4I5 , 3F10.0)
  208 FORMAT('OLOADING' /    ' CONSTANT COEFFICIENT',T40, F13.3)
  209 FORMAT(' LINEAR IN X', T40, F13.3, 'X')
  210 FORMAT(' LINEAR IN Y', T40, F13.3, 'Y')
  211 FORMAT(' SQUARE IN X', T40, F13.3, 'X*X')
  212 FORMAT(' SQUARE IN Y', T40, F13.3, 'Y*Y')
  213 FORMAT(' CROSS TERM ', T40, F13.3, 'X*Y')
  214 FORMAT(' FULL SINE  ', T40, F13.3,
     1    'SIN(PHI*X/A) * SIN(PHI*Y/B)'        )
  215 FORMAT(' HALF SINE  ', T40, F13.3,
     1    'SIN(PHI*X/2*A)*SIN(PHI*Y/2*B)'    )
  216 FORMAT('ODISPLACEMENT BOUNDARY CONDITIONS' /
     1     '  ICODE    X-INDEX    Y-INDEX    Z-INDEX    X-MAGNITUDE    Y-MAGN
```

```
  2ITUDE   Z-MAGNITUDE'          )
217 FORMAT(I5,T13,I2,T23,I2,T33,I2,T41,G11.3,T55,G11.3,T69,G11.3)
218 FORMAT('OMATERIAL PROPERIES AT NODES' /
  1      '  MTLND    X-INDEX    Y-INDEX    Z-INDEX    ')
    DO 75 I=1,78
 75 VLDMAT(I) = RELVT(I)
    READ(ICRD,102)  ZLCOEF
    READ(ICRD,102)      BCTMP
    READ(ICRD,103)     NMTLCD
    WRITE(IWRT,208)        ZLCOEF(1)
    WRITE(IWRT,209)        ZLCOEF(2)
    WRITE(IWRT,210)        ZLCOEF(3)
    WRITE(IWRT,211)        ZLCOEF(4)
    WRITE(IWRT,212)        ZLCOEF(5)
    WRITE(IWRT,213)        ZLCOEF(6)
    WRITE(IWRT,214)        ZLCOEF(7)
    WRITE(IWRT,215)        ZLCOEF(8)
    LU = 2
    DO 42 I=1,NGNP
    ICODE(I) = 0
    UX(I) = 0D0
    UY(I) = 0D0
    UZ(I) = 0D0
 42 P(I) = 0D0
    NMNN1 = NEL-NELY*NELX+1
    DO 45 INEL=NMNN1,NEL
    DO 43 I=1,12
    L = LDRLVT(I,LU)
    T = X(IX(INEL,L))
    S = Y(IX(INEL,L))
 43 PBAR(I)              = ZLCOEF(1) + ZLCOEF(2)*T + ZLCOEF(3)*S
  1      + ZLCOEF(4)*T*T + ZLCOEF(5)*S*S + ZLCOEF(6)*T*S
  2     +ZLCOEF(7)*DSIN(3.14159*T/AX) *DSIN(3.14159*S/BY)
  3     + ZLCOEF(8) *DSIN(3.14159*T/(2*AX)) *DSIN(3.14159*S/(2*BY))
    M = 0
    DO 44 I=1,12
    K = IX(INEL,LDRLVT(I,LU))
    DO 44 J=I,12
    L = IX(INEL,LDRLVT(J,LU))
    M = M + 1
    P(K) = P(K) + VLDMAT(M) * PBAR(J)
    IF(I .EQ. J)  GO TO 44
    P(L) = P(L) + VLDMAT(M) * PBAR(I)
 44 CONTINUE
 45 CONTINUE
    ELLY = BY/NELY
    ELLX = AX/NELX
    AB = ELLX*ELLY
    DO 46 IGNP=1, NGNP
 46 UZ(IGNP)= AB *P(IGNP) / 100800.
    IF(NMTLCD .EQ. 0 )  GO TO 41
```

```
      CALL TITLE
      WRITE(IWRT,218)
      DO 40 IMTLCD=1,NMTLCD
      READ(ICRD,103)  MCODE, IXC, IYC, IZC
      WRITE(IWRT,217) MCODE, IXC, IYC, IZC
      IF(IXC .NE. 0 )  GO TO 78
      IXO = 1
      IXF = 3*NELX+1
      GO TO 79
   78 IXO = IXC
      IXF = IXC
   79 IF(IYC .NE. 0 )  GO TO 80
      IYO = 1
      IYF = 3*NELY+1
      GO TO 81
   80 IYO = IYC
      IYF = IYC
   81 IF(IZC .NE. 0 )  GO TO 82
      IZO = 1
      IZF = NELZ+1
      GO TO 83
   82 IZO = IZC
      IZF = IZC
   83 CONTINUE
      DO 86 I=IXO,IXF
      DO 86 J=IYO,IYF
      DO 86 K=IZO,IZF
      IF(IDPIX1(I,J,K) .EQ. C )  GO TO 86
      MTLND(IDPIX1(I,J,K)) = MCODE
   86 CONTINUE
   40 CONTINUE
   41 CALL TITLE
      WRITE(IWRT,216)
   47 READ(ICRD,103,END=58)LCODE,IXC,IYC,IZC,DBCX,DBCY,DBCZ
      WRITE(IWRT,217)   .   LCODE, IXC, IYC, IZC, DBCX, DBCY, DBCZ
      IF(IXC .NE. 0 )  GO TO 48
      IXO = 1
      IXF = 3*NELX+1
      GO TO 49
   48 IXO = IXC
      IXF = IXC
   49 IF(IYC .NE. 0 )  GO TO 50
      IYO = 1
      IYF = 3*NELY+1
      GO TO 51
   50 IYO = IYC
      IYF = IYC
   51 IF(IZC .NE. 0 )  GO TO 52
      IZO = 1
      IZF = NELZ+1
      GO TO 53
```

```
52 IZO = IZC                                              BCP 1540
   IZF = IZC                                              BCP 1550
53 CONTINUE                                               BCP 1560
   DO 56 I=IXO,IXF                                        BCP 1570
   DO 56 J=IYO,IYF                                        BCP 1580
   DO 56 K=IZO,IZF                                        BCP 1590
   IF(IDPIX1(I,J,K) .EQ. 0 )  GO TO 56                    BCP 1600
   ICODE(IDPIX1(I,J,K)) = LCODE                           BCP 1610
   IF(LCODE .EQ. 0)  GO TO 56                             BCP 1620
   IF(LCODE .EQ. 4  .OR.  LCODE .EQ. 6)  GO TO 54         BCP 1630
   IF(LCODE .EQ. 5)  GO TO 55                             BCP 1640
   UX(IDPIX1(I,J,K)) = DBCX                               BCP 1650
   IF(LCODE .EQ. 1)  GO TO 56                             BCP 1660
   IF(LCODE .EQ. 3)  GO TO 55                             BCP 1670
54 UY(IDPIX1(I,J,K)) = DBCY                               BCP 1680
   IF(LCODE .EQ. 2  .OR.  LCODE .EQ. 4)  GO TO 56         BCP 1690
55 UZ(IDPIX1(I,J,K)) = DBCZ                               BCP 1700
56 CONTINUE                                               BCP 1710
   GO TO 47                                               BCP 1720
58 RETURN                                                 BCP 1730
   END                                                    BCP 1740
```

```
      SUBROUTINE  DISPLY
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   *  SUBROUTINE DISPLY DISPLAYS THE MESH                            *
C   *                                                                 *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT RE'  8 (A-H,O-Z)
      DOUBLE PRE( .ION  NUMB(6)
      INTEGER*2   IX, ICODE, IDPIX1, MTLND
      INTEGER*2   IDPL1(19)
      COMMON /GENMA1/ X(1066),Y(1066),Z(1066),UX(1066),UY(1066),UZ(1066)
     1 ,  TMPND(1066), BCTMP(8),
     2 ALFA1(10),ALFA2(10),ALFA3(10)   'ORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE 1C    ,IDPIX1(19,19,11),MTLND(1066)
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1    NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2    , ITYTD, NELX31, NELY31, NELZ1
      COMMON /DATDIS/ XCRD(19), YCRD(19), ZCRD(11)
      COMMON / HEAD /  HED(10),IC    WRT,IPAGE,LINE
      DIMENSION    VERTLN(19),  EL   (6)
      DATA  VERTLN  / 6H      |, 6H      |, 6H      |, 6H      |, 6H      |,
     1               6H      |, 6H      |, 6H      |, 6H      |, 6H      |,
     2               6H      |, 6H      |, 6H      |, 6H      |, 6H      |,
     3               6H      |, 6H      |, 6H      |, 6H      |    /
      DATA ELEMT / 7HELEMENT, 7HELEMENT, 7HELEMENT, 7HELEMENT,
     1            7HELEMENT, 7HELEMENT  /
      DATA NUMB   / 7HNUMBERS,7HNUMBERS,7HNUMBERS,7HNUMBERS,7HNUMBERS,
     1            7HNUMBERS /
  199 FORMAT('0   Z-COORD =' , E14.7)
  200 FORMAT('0', 14X, 19F6.2)
  201 FORMAT(12X,19A6)
  202 FORMAT(F11.2, ' -- ', 19(I4,'--'))
  203 FORMAT(12X, A6, 6(12X, A6))
  204 FORMAT(F11.2, ' -- ', I4, 6(4X,A7,I7))
  205 FORMAT(17X, '|', 6(5X, A7,   A6))
  206 FORMAT(F11.2,' -- ', I4, 18(I6))
      IWRT = 6
      M=0
      DO 38 I=1,NELX
      DO 38 L=1,NELY
      M=M+1
   38 IDPIX1(3*I   ,3*L-1, 1   ) = M
      IF(NELZ .LT. 2)  GO TO 40
      DO 39 IELZ=2,NELZ
      DO 39 I=1,NELX
      DO 39 L=1,NELY
      M = M + 1
      IDPIX1(3*I   ,3*L-1,IELZ) = M
   39 IDPIX1(3*I-1,3*L-1,IELZ) = IDPIX1(3*I   ,3*L-1,IELZ-1)
```

```
40 DO 41 I=1,NELX
   DO 41 L=1,NELY
   M = M + 1
41 IDPIX1(3*I-1,3*L-1,NELZ+1)=IDPIX1(3*I   ,3*L-1,NELZ   )
   NELX31 = NELX*3+1
   NELY31 = NELY*3+1
   IELZ = 0
11 NELX6 = NELX
   ISW1 = 0
   IELZ = IELZ+1
   CALL TITLE
   WRITE(IWRT,199)  ZCRD(IELZ)
 9 IF(NELX6 .GT. 6 )  GO TO 8
   NELX36 = 3*NELX6+1
   NELX16= NELX6+1
   NELX26 = 2*NELX6+1
   NELX62= 2*NELX6
   NELX60 = NELX6
   GO TO 12
 8 NELX36 = 19
   NELX16 = 7
   NELX26 = 13
   NELX62 = 12
   NELX60 = 6
12 IF(ISW1 .EQ. 0)  GO TO 13
   N1 = 3*(NELX6+6)+1
   DO 10 I=19,N1
   XCRD(I-18) = XCRD(I)
   DO 10 J=1,NELY31
10 IDPIX1(I-18,J,IELZ) = IDPIX1(I,J,IELZ)
13 ISW1 = 1
   WRITE(IWRT,200)    (XCRD(J),J=1,NELX36)
   WRITE(IWRT,201)  (VERTLN(J),J=1,NELX36)
   WRITE(IWRT,201)  (VERTLN(J),J=1,NELX36)
   WRITE(IWRT,202) YCRD(NELY31) , (IDPIX1(J,NELY31,IELZ),J=1,NELX36)
   DO 35 I=1,NELY
   NELY3I = 3*(NELY-I)
   WRITE(IWRT,203)  (VERTLN(J),J=1,NELX16)
   WRITE(IWRT,203)  (VERTLN(J),J=1,NELX16)
   M = 0
   DO 45 L=1,NELX36,3
   M = M+1
45 IDPL1(M) = IDPIX1(L,NELY3I+3, IELZ)
   WRITE(IWRT,204)    YCRD(NELY3I+3) , (IDPL1(J),ELEMT(J),J=1,NELX60),
  1         IDPL1(NELX60+1)
   WRITE(IWRT,205)    (NUMB(J), VERTLN(J),J=1,NELX60)
   WRITE(IWRT,203)  (VERTLN(J),J=1,NELX16)
   WRITE(IWRT,206) YCRD(NELY3I+2) , (IDPIX1(J,NELY3I+2,IELZ),
  1       J=1,NELX36)
   WRITE(IWRT,203)  (VERTLN(J),J=1,NELX16)
   WRITE(IWRT,203)  (VERTLN(J),J=1,NELX16)
```

```
35 WRITE(IWRT,202)YCRD(NELY3I+1),(IDPIX1(J,NELY3I+1,IELZ),J=1,NELX36)DIP 1030
   NELX6 = NELX6-6                                                  DIP 1040
   IF(NELX6 .LT. 1)  GO TO 7                                        DIP 1050
   GO TO 9                                                          DIP 1060
 7 IF(IELZ .LE. NELZ  )  GO TO 11                                   DIP 1070
   RETURN                                                           DIP 1080
   END                                                              DIP 1090
```

```
      SUBROUTINE TITLE
C
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    *                                                               *
C    *  SUBROUTINE TITLE PRINTS THE HEADING ON EACH PAGE             *
C    *                                                               *
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
  100 FORMAT (1H1,'FEM 72-DOF GENERAL HEXAHEDRONS THERMO-ELASTIC, VARYIN
     1G MATERIAL PROPERTIES, DANA', 9X, 'PAGE', I3)
  101 FORMAT (1H0,10A8 )
      WRITE (IWRT,100) IPAGE
      WRITE (IWRT,101) HED
      IPAGE= IPAGE +1
      LINE = 0
      RETURN
      END
```

```
      SUBROUTINE  TMPDST
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE TMPDST SOLVES FOR ONE-DIMENSIONAL TEMPERATURE         *
C   *  DISTRIBUTIONS                                                    *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, ICODE, IDPIX1, MTLND
      COMMON /GENMAT/ X(1066),Y(1066),Z(1066),UX(1066),UY(1066),UZ(1066)
     1 ,  TMPND(1066), BCTMP(8),
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1066),IDPIX1(19,19,11),MTLND(1066)
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1    NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2   , ITYTD, NELX31, NELY31, NELZ1
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
      COMMON / TMPRTR/   TEMP1(19), TEMP2(19,19), TEMP3(19,19,11)
  200 FORMAT('0 THE TEMPERATURE DISTRIBUTION IS CONSTANT AT',
     1    E15.7, ' DEGREES CENT.'  )
  201 FORMAT('0 THE TEMPERATURE DISTRIBUTION VARIES FROM' , E15.7,
     1    ' TO', E15.7, ' IN THE X-DIRECTION ONLY'  )
  202 FORMAT('0 THE TEMPERATURE DISTRIBUTION VARIES FROM' , E15.7,
     1    ' TO', E15.7, ' IN THE Y-DIRECTION ONLY'  )
  203 FORMAT('0 THE TEMPERATURE DISTRIBUTION VARIES FROM' , E15.7,
     1    ' TO', E15.7, ' IN THE Z-DIRECTION ONLY'  )
  204 FORMAT('0 TEMPERATURE DISTRIBUTION VARIES IN THE X-Y PLANE ONLY'/
     1    '0 TEMPERATURE BC1 = ', E15.7/
     1    '  TEMPERATURE BC2 = ', E15.7/
     1    '  TEMPERATURE BC3 = ', E15.7/
     1    '  TEMPERATURE BC4 = ', E15.7)
  205 FORMAT('0 TEMPERATURE DISTRIBUTION VARIES IN THE X-Z PLANE ONLY'/
     1    '0 TEMPERATURE BC1 = ', E15.7/
     1    '  TEMPERATURE BC2 = ', E15.7/
     1    '  TEMPERATURE BC5 = ', E15.7/
     1    '  TEMPERATURE BC6 = ', E15.7)
  206 FORMAT('0 TEMPERATURE DISTRIBUTION VARIES IN THE Y-Z PLANE ONLY'/
     1    '0 TEMPERATURE BC1 = ', E15.7/
     1    '  TEMPERATURE BC3 = ', E15.7/
     1    '  TEMPERATURE BC5 = ', E15.7/
     1    '  TEMPERATURE BC7 = ', E15.7)
  207 FORMAT('0 3-D TEMPERATURE DISTRIBUTION SUBROUTINE NOT COMPLETE')
      IF (ITYTD .NE.  0  )  GO TO 10
      WRITE(IWRT,200)  BCTMP(1)
      DO 20 I=1,NGNP
   20 TMPND(I) = BCTMP(1)
      GO TO 9
   10 NICMIX = NELX31 - 1
      NICMIY = NELY31 - 1
```

```
      GO TO (1, 2, 3, 4, 5, 6, 7), ITYTD
    1 TMPINC = (BCTMP(2) - BCTMP(1) ) / NICMTX
      WRITE(IWRT,201)  BCTMP(1), BCTMP(2)
      DO 21 I=1,NICMTX
   21 TEMP1(I) = BCTMP(1) + TMPINC*(I-1)
      TEMP1(NELX31) = BCTMP(2)
      DO 31 I=1,NELX31
      DO 31 J=1,NELY31
      DO 31 K=1,NELZ1
   31 TEMP3(I,J,K) = TEMP1(I)
      GO TO 8
    2 TMPINC = (BCTMP(3) - BCTMP(1) ) / NICMTY
      WRITE(IWRT,202)  BCTMP(1), BCTMP(3)
      DO 22 J=1,NICMTY
   22 TEMP1(J) = BCTMP(1) + TMPINC*(J-1)
      TEMP1(NELY31) = BCTMP(3)
      DO 32 I=1,NELX31
      DO 32 J=1,NELY31
      DO 32 K=1,NELZ1
   32 TEMP3(I,J,K) = TEMP1(J)
      GO TO 8
    3 TMPINC = (BCTMP(5) - BCTMP(1) ) / NELZ
      WRITE(IWRT,203)  BCTMP(1), BCTMP(5)
      DO 23 K=1,NELZ
   23 TEMP1(K) = BCTMP(1) + TMPINC*(K-1)
      TEMP1(NELZ+1) = BCTMP(5)
      DO 33 I=1,NELX31
      DO 33 J=1,NELY31
      DO 33 K=1,NELZ1
   33 TEMP3(I,J,K) = TEMP1(K)
      GO TO 8
    4 WRITE(IWRT,204)  BCTMP(1), BCTMP(2), BCTMP(3), BCTMP(4)
      BC1 =BCTMP(1)
      BC2 =BCTMP(2)
      BC3 =BCTMP(3)
      BC4 =BCTMP(4)
      CALL TWOD( THELLX, THELLY, NELX31, NELY31, BC1, BC2, BC3, BC4)
      DO 34 I=1,NELX31
      DO 34 J=1,NELY31
      DO 34 K=1,NELZ1
   34 TEMP3(I,J,K) = TEMP2(I,J)
      GO TO 8
    5 WRITE(IWRT,205)  BCTMP(1), BCTMP(2), BCTMP(5), BCTMP(6)
      BC1 =BCTMP(1)
      BC2 =BCTMP(2)
      BC3 =BCTMP(5)
      BC4 =BCTMP(6)
      CALL TWOD( THELLX,   ELLZ, NELX31, NELZ1 , BC1, BC2, BC3, BC4)
      DO 35 I=1,NELX31
      DO 35 J=1,NELY31
      DO 35 K=1,NELZ1
```

```
35 TEMP3(I,J,K) = TEMP2(J,K)
   GO TO 8
 6 WRITE(IWRT,206)  BCTMP(1), BCTMP(3), BCTMP(5), BCTMP(7)
   BC1 =BCTMP(1)
   BC2 =BCTMP(3)
   BC3 =BCTMP(5)
   BC4 =BCTMP(7)
   CALL TWOD( THELLY,   ELLZ, NELY31, NELZ1 , BC1, BC2, BC3, BC4)
   DO 36 I=1,NELX31
   DO 36 J=1,NELY31
   DO 36 K=1,NELZ1
36 TEMP3(I,J,K) = TEMP2(J,K)
   GO TO 8
 7 WRITE(IWRT,207)
 8 DO 40 I=1,NELX31
   DO 40 J=1,NELY31
   DO 40 K=1,NELZ1
   IF(IDPIX1(I,J,K)  .EQ.  0)  GO TO 40
   TMPND(IDPIX1(I,J,K)) = TEMP3(I,J,K)
40 CONTINUE
 9 RETURN
   END
```

```
      SUBROUTINE  TWOD(H1, H2, NPTS1, NPTS2, BC1, BC2, BC3, BC4)
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE TWOD SOLVES FOR TWO-DIMENSIONAL TEMPERATURE           *
C   *  DISTRUBUTIONS                                                    *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1    NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2   , ITYTD, NELX31, NELY31, NELZ1
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
      COMMON / TMPRTR/   TEMP1(19), TEMP2(19,19), TEMP3(19,19,11)
 1000 FORMAT( '0 TEMPERATURE DISTRIBUTION CONVERGED TO',  E15.7,
     1   ' IN', I5, '  ITERATIONS'  )
 1001 FORMAT( '0 TEMPERATURE DISTRIBUTION DID NOT CONVERGE TO',E15.7,
     1   '  IN', I5, '  ITERATIONS'  )
      NICMT1 = NPTS1 - 1
      NICMT2 = NPTS2 - 1
      DO 10 I=1,NPTS1
      DO 10 J=1,NPTS2
   10 TEMP2(I,J) = TMPINT
      TMPINC = (BC2 - BC1) / NICMT1
      DO 11 I=2,NICMT1
   11 TEMP2(I,1) = BC1 + TMPINC*(I-1)
      TMPINC = (BC4 - BC3) / NICMT1
      DO 12 I=1,NICMT1
   12 TEMP2(I,NPTS2)     = BC3 + TMPINC*(I-1)
      TMPINC = (BC3 - BC1) / NICMT2
      DO 13 J=1,NICMT2
   13 TEMP2(1,J) = BC1 + TMPINC*(J-1)
      TMPINC = (BC4 - BC2) / NICMT2
      DO 14 J=1,NICMT2
   14 TEMP2(NPTS1,J)     = BC2 + TMPINC*(J-1)
      TEMP2(NPTS1,NPTS2) = BC4
      NOCVT = 0
      KOUNT= 0
      H1SQ = H1*H1
      H2SQ = H2*H2
      DEM = 2.D0 * (1.D0/H1SQ + 1.D0/H2SQ)
    1 IF( KOUNT .GT. LMTMP)  GO TO 2
      ERRMAX = 0.D0
      KOUNT = KOUNT + 1
      DO 20 I=2,NICMT1
      DO 20 J=2,NICMT2
      OLDTMP = TEMP2(I,J)
      TEMP2(I,J) = ((TEMP2(I-1,J) + TEMP2(I+1,J)) / H1SQ
     1    +     (TEMP2(I,J-1) + TEMP2(I,J+1)  ) / H2SQ )    /  DEM
      ERR = DABS( OLDTMP - TEMP2(I,J) )
```

```
      IF( ERR  .GT. ERRMAX )  ERRMAX = ERR
   20 CONTINUE
      IF( ERRMAX  .GT. EPSTMP ) GO TO 1
      WRITE(IWRT,1000)  ERRMAX,  KOUNT
      RETURN
    2 WRITE(IWRT,1001)  EPSTMP,  KOUNT
      NOCVT = 1
      RETURN
      END
```

```
      SUBROUTINE MODF
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                 *
C     *  SUBROUTINE MODF DOES NOTHING — THIS SUBROUTINE CAN BE USED TO *
C     *  MODIFY ANY INFORMATION THAT HAS BEEN GENERATED                 *
C     *                                                                 *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2   IX, ICODE, IDPIX1, MTLND
      COMMON /GENMAT/ X(1066),Y(1066),Z(1066),UX(1066),UY(1066),UZ(1066)
     1 ;  TMPND(1066), BCTMP(8),
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1066),IDPIX1(19,19,11),MTLND(1066)
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1     NEL, NGNP, NGLOF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2   , ITYTO, NELX31, NELY31, NELZ1
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
      WRITE(IWRT,200)
  200 FORMAT(// '0  DATA GENERATION STEP HAS NOT BEEN MODIFIED' )
      RETURN
      END
```

APPENDIX D

Hole in Rectangular Plate Mesh Generator

A. Introduction

This mesh generator will yield element, nodal, and material data necessary to idealize a laminated composite pierced with a hole and subjected to axial and thermal loads. The shape of the hole can be circular, square or diamond. Loads are applied as a result of a uniform axial displacement in the x-direction at $x = \pm a$. The thermal effects are restricted to a constant temperature change. The mesh is restricted to the shape shown in Figure D-1 where a, b, t, c, e and R can be varied. The number of elements through-the-thickness is also a variable.

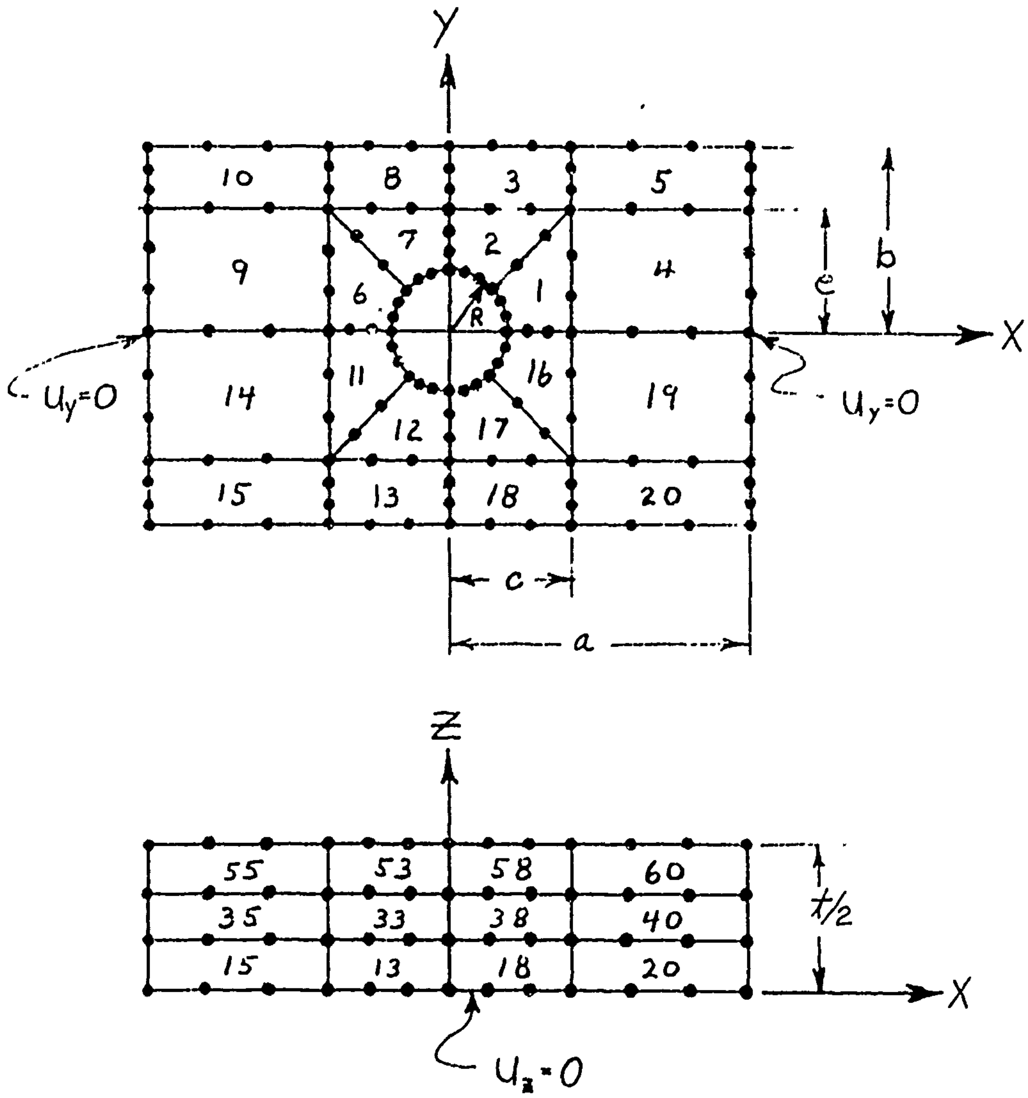

FIGURE D-1: Mesh for Rectangular Plate with a Hole

B. Input Data

1. Heading card (10A8)

   Columns 1-80 information to be printed with output

2. Output unit card (I5) one card

   Columns 1-5 unit number for passing data to next job step

3. Control and problem parameter card (I10, 2F10.0) one card

   Columns 1-10 type of thermal-elastic problem

   11-20 initial temperature

   21-30 final temperatures

4. Plate and mesh dimensions cards two cards

   First card (3 G10.0, I5)

   Columns 1-10 a-dimension, inches

   11-20 b-dimension, inches

   21-30 t/2-dimension, inches

   30-35 number of elements through the half thickness

   Second card (3G10.1)

   Columns 1-10 c-dimension, inches

   11-20 e-dimension, inches

   21-30 R-dimension, inches

5. Load and hole parameters (3I5, F10.0) one card

   Columns 1-5 '0' if hole open, '1' if hole filled

   6-10 '1' if circular hole

   '2' if square hole

   '3' if diamond hole

   11-15 blank

   16-26 magnitude of displacement, inches

6. Material change data cards (16I5)

   Columns 1-5 number of materials

   5-10 number of material changes

   11-15 material number

   16-20 element number at which the material changed

   (Use as many sets of material number and element

number as required to describe at which element a material is changed. The elements are numbered, on the plate, as shown in Figure D-1.)

7. Material data cards two cards

First card (2I5, F10.2, 3F10.8) one for each material

Columns 1-5 material number (in sequential order)

6-10 number of material cards ('1' for class 1, 2 or 3)

11-20 fiber orientation in degrees

21-30 thermal expansion coefficient, α_{11}

31-40 thermal expansion coefficient, α_{22}

41-50 thermal expansion coefficient, α_{33}

Subsequent cards (F5.0, 3F10.0, 3F5.2, 3F10.0) (One card for problem class 1, 2 or 3. And for problem class 4, one card for each temperature for which material properties are specified.)

Columns 1-5 temperature for material properties (can be left blank for class 1 and 2 problems)

6-15 modulus of elasticity, E_{11}, KSI

16-25 modulus of elasticity, E_{22}, KSI

26-35 modulus of elasticity, E_{33}, KSI

36-40 Poisson's ratio, ν_{12}

41-45 Poisson's ratio, ν_{13}

46-50 Poisson's ratio, ν_{23}

51-60 shear modulus, G_{12}, KSI

61-70 shear modulus, G_{13}, KSI

71-80 shear modulus, G_{23}, KSI

8. Element change data cards (16I5)

Columns 1-5 number of unique elements

6-10 number of element changes

11-15 element type number

16-20 element number at which the element type changes (Use as many sets of element type and element number as required to describe at which element number an element type is changed.)

```
C        MAIN PROGRAM STEP 1B
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                   *
C     *  THIS MESH GENERATOR IDEALIZES A RECTANGULAR LAMINATED PLATE       *
C     *  WITH A HOLE, LOADED IN UNIFORM EXTENSION IN THE X-DIRECTION      *
C     *                                                                   *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2    IX, ICODE, IDPIX1, MTLND
      COMMON /GENMAT/ X(1015),Y(1015),Z(1015),UX(1015),UY(1015),UZ(1015)
     1 ,   TMPND(1015), BCTMP,
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1015),IDPIX1(19,19,11),MTLND(1015)
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1     NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASS
     2    , ITYTD, NELX31, NELY31, NELZ1
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
 1000 FORMAT( 16I5)
 1001 FORMAT(2I5,  F10.2, 3F10.8)
 1002 FORMAT(F5.0, 3F10.0, 3F5.2, 3F10.0)
 1003 FORMAT( I4, I4, I2, 6F10.6, F10.2)
 1004 FORMAT( 10A8)
 1005 FORMAT( 4I5, F10.2)
 1011 FORMAT(I10, 2F10.5)
      ICRD = 5
      IPAGE = 1
      IWRT = 6
      READ(5,1004)   HED
      READ(5,1000)   NTUT
      READ(5,1011)   ICLASS, AMBTMP, BCTMP
C
C   GENERATE MESH AND BOUNDARY CONDITIONS
C
      CALL DATGEN
      DO 20 I=1,NGNP
   20 TMPND(I) = BCTMP
      CALL MODF
C
C   WRITE MESH DATA ON UNIT NTUT
C
      WRITE(NTUT,1004) HED
      WRITE(NTUT,1005)  NGNP, NMTL, NEL, NTYEL, AMBTMP
      DO 10 IMTL=1,NMTL
      WRITE(NTUT,1001)  IMTL, NTMP(IMTL), FIBORT(IMTL), ALFA1(IMTL),
     1        ALFA2(IMTL), ALFA3(IMTL)
      NTMP1 = NTMP(IMTL)
      DO 10 ITMP=1,NTMP1
   10 WRITE(NTUT,1002)   TMPEL(IMTL,ITMP), (E(IMTL,J,ITMP),J=1,9)
      DO 30 INEL=1,NEL
```

```
30 WRITE(NTUT,1000)  INEL,  (IX(INEL,J),J=1,27)
   DO 40    M=1,NGNP
40 WRITE(NTUT,1003)  M, MTLND(M), ICODE(M), X(M), Y(M), Z(M),
  1     UX(M), UY(M), UZ(M), TMPND(M)
   STOP
   END
```

```
      SUBROUTINE DATGEN
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                 *
C     *  SUBROUTINE DATGEN GENERATES THE MESH, NUMBERS THE NODES AND    *
C     *  ELEMENTS, AND SPECIFIES BOUNDARY CONDITION CODES FOR EACH NODE *
C     *                                                                 *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2    IX, ICODE, IDPIX1, MTLND
      COMMON /GENMAT/ X(1015),Y(1015),Z(1015),UX(1015),UY(1015),UZ(1015)
     1 ,  TMPND(1C15), BCTMP,
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(1C),E(10,9,10),TMPEL(10,10),
     3  NTMP(10),IX(144,27),ICODE(1015),IDPIX1(19,19,11),MTLND(1015)
      COMMON  / GENL1 /  TMPINI, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,
     1     NEL, NGNP, NGLDF, NMTL, NTYLL, LMTMP, NELX, NELY, NELZ, ICLASS
     2    , ITYTD, NELX31, NELY31, NELZ1
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
      DIMENSION
     1      IXFLCH(145), IXMLCH(145), IMATL(145), ITYEL(145)
      DIMENSION  IXDT1(24), IXDT2(24), IXDT3(24), IXDT4(24), IXDT5(24)
      DIMENSION  IXDT6(24), IXDT7(24), IXDT8(24), IXDT9(24), IXDT10(24)
      DIMENSION   IXDT11(24),IXDT12(24),IXDT13(24),IXDT14(24),IXDT15(24)
      DIMENSION  IXDT16(24),IXDT17(24),IXDT18(24),IXDT19(24),IXDT20(24)
      DIMENSION   IXDTP1(24), IXDTP2(24), IXDTP3(24), IXDTP4(24)
      DIMENSION   IPLNP1(10), IPLNP2(10), IPLNP3(10), IPLNP4(10)
      DIMENSION  IDX1(10), IDX2(10), IDX3(10), IDX4(10)
      DIMENSION   XLDSQ(4)
      DATA IXDT1/1, 14, 18, 22, 137, 150, 154, 158, 2, 3, 23, 24, 138,
     1  139, 159, 160, 4, 15, 19, 25, 140, 151, 155, 161 /
      DATA IXDT2/ 7, 6, 5, 4, 143, 142, 141, 140, 8, 9, 15, 19, 144,
     1  145, 151, 155, 10, 16, 20, 25, 146, 152, 156, 161 /
      DATA IXDT3/ 10, 16, 20, 25, 146, 152, 156, 161, 11, 12, 26, 27,
     1  147, 148, 162, 163, 13, 17, 21, 28, 149, 153, 157, 164 /
      DATA IXDT4/ 22, 29, 32, 35, 158, 165, 168, 171, 23, 24, 36, 37,
     1  159, 160, 172, 173, 25, 30, 33, 38, 161, 166, 169, 174 /
      DATA IXDT5/ 25, 30, 33, 38, 161, 166, 169, 174, 26, 27, 39, 40,
     1  162, 163, 175, 176, 28, 31, 34, 41, 164, 167, 170, 177 /
      DATA IXDT6/ 56, 52, 48, 42, 192, 188, 184, 178, 57, 58, 43, 44,
     1  193, 194, 179, 180, 59, 53, 49, 45, 195, 189, 185, 181 /
      DATA IXDT7/ 45, 46, 47, 7, 181, 182, 183, 143, 49, 53, 8, 9,
     1  185, 189, 144, 145, 59, 54, 50, 10, 195, 190, 186, 146 /
      DATA IXDT8/ 59, 54, 50, 10, 195, 190, 186, 146, 60, 61, 11, 12,
     1  196, 197, 147, 148, 62, 55, 51, 13, 198, 191, 187, 149 /
      DATA IXDT9/ 69, 66, 63, 56, 205, 202, 199, 192, 70, 71, 57, 58,
     1  206, 207, 193, 194, 72, 67, 64, 59, 208, 203, 200, 195 /
      DATAIXDT10/ 72, 67, 64, 59, 208, 203, 200, 195, 73, 74, 60, 61,
     1  209, 210, 196, 197, 75, 68, 65, 62, 211, 204, 201, 198 /
      DATA IXDT11/ 96, 91, 88, 78, 232, 227, 224, 214, 95, 94, 77, 76,
     1  231, 230, 213, 212, 56, 52, 48, 42, 192, 188, 184, 178 /
```

```
      DATA IXDT12/ 96, 92, 89, 84, 232, 228, 225, 220, 91, 88, 83, 82,
     1 227, 224, 219, 218, 78, 79, 80, 81, 214, 215, 216, 217 /
      DATA IXDT13/ 99, 93, 90, 87, 235, 229, 226, 223, 98, 97, 86, 85,
     1 234, 233, 222, 221, 96, 92, 89, 84, 232, 228, 225, 220 /
      DATA IXDT14/ 106, 102, 100, 96, 242, 238, 236, 232, 105, 104, 95,
     1 94, 241, 240, 231, 230, 69, 66, 63, 56, 205, 202, 199, 192 /
      DATA IXDT15/ 109, 103, 101, 99, 245, 239, 237, 235, 108, 107, 98,
     1 97, 244, 243, 234, 233, 106, 102, 100, 96, 242, 238, 236, 232 /
      DATA IXDT16/ 112, 115, 118, 123, 248, 251, 254, 259, 111, 110,
     1 122, 121, 247, 246, 258, 257, 1, 14, 18, 22, 137, 150, 154, 158/
      DATA IXDT17/ 84, 116, 119, 123, 220, 252, 255, 259, 83, 82, 118,
     1 115, 219, 218, 254, 251, 81, 114, 113, 112, 217, 250, 249, 248/
      DATA IXDT18/ 87, 117, 120, 126, 223, 253, 256, 262, 86, 85, 125,
     1 124, 222, 221, 261, 260, 84, 116, 119, 123, 220, 252, 255, 259 /
      DATA IXDT19/ 123, 127, 129, 133, 259, 263, 265, 269, 122, 121,132,
     1 131, 258, 257, 268, 267, 22, 29, 32, 35, 158, 165, 168, 171 /
      DATA IXDT20/ 126, 128, 130, 136, 262, 264, 266, 272, 125, 124,135,
     1 134, 261, 260, 271, 270, 123, 127, 129, 133, 259, 263, 265, 269/
      DATA IXDTP1 / 0,0,0,1,0,0,0,137,0,0,2,3,0,0,138,139,
     1         7,6,5,4,143,142,141,140 /
      DATA IXDTP2 / 42,0,0,0,178,0,0,0,43,44,0,0,179,180,0,0,
     1         45,46,47,7,181,182,183,143 /
      DATA IXDTP3 / 78,79,80,81,214,215,216,217,77,76,0,0,213,212,0,0,
     1         42,0,0,0,178,0,0,0 /
      DATA IXDTP4 / 81,114,113,112,217,250,249,248,0,0,111,110,0,0,247,
     1         246,0,0,0,1,0,0,0,137 /
      DATA IDX1 / 1,2,3,5,6,7,9,10,13,14 /
      DATA IDX2 / 2,3,4,6,7,8,11,12,15,16 /
      DATA IDX3 / 11,12,15,16,18,19,20,22,23,24 /
      DATA IDX4 / 9,10,13,14,17,18,19,21,22,23 /
      DATA IPLNP1 / 3,4,5,12,13,14,7,6,16,15 /
      DATA IPLNP2 / 1,2,3,10,11,12,7,6,16,15 /
      DATA IPLNP3 / 9,8,18,17,1,2,3,10,11,12 /
      DATA IPLNP4 / 9,8,18,17,3,4,5,12,13,14 /
      DATA    XLDSQ  /  1.D0, 3.D0, 3.D0, 1.D0 /
  100 FORMAT(16I5)
  102 FORMAT( 3G10.0,I5)
  103 FORMAT(3G10.1)
  104 FORMAT(3I5, G10.0)
  202 FORMAT('0LENGTH OF STRIP'                     ,T50,G24.7 /
     1      '0WIDTH OF STRIP'                      ,T50,G24.7 /
     2      '0THICKNESS OF STRIP'                  ,T50,G24.7 /
     3      '0NUMBER OF ELEMENTS THICK'            ,T50,G24.7 /
     4      '0LENGTH OF INSERT'                    ,T50,G24.7 /
     5      '0WIDTH OF INSERT'                     ,T50,G24.7 /
     6      '0MAXIMUM WIDTH OF HOLE'               ,T50,G24.7 )
  203 FORMAT('0UNIFORM DISPLACEMENT IN INCHES OF '    , T50, G24.7)
  204 FORMAT('0UNIFORM LOAD IN KIPS OF '  ,T50,  G24.7)
  205 FORMAT('0HOLE IS FILLED')
  206 FORMAT('0MATERIAL TYPE AND MATERIAL CHANGES'  /
     1   T10,  'MATERIAL TYPE          CHANGE AT ELEMENT'  )
```

```
  207 FORMAT('OCIRCULAR HOLE' )
  208 FORMAT('OSQUARE HOLE' )
  209 FORMAT('ODIAMOND HOLE' )
  307 FORMAT('CELEMENT TYPE AND ELEMENT CHANGES'  /
     1   T10,  'ELEMENT TYPE           CHANGE AT ELEMENT'  )
  306 FORMAT(18X, I2, 22X, I4)
 1001 FORMAT(2I5,  F10.2, 3F10.8)
 1002 FORMAT(F5.0, 3F10.0, 3F5.2, 3F10.0)
      READ(ICRD,102)    XLNTH, YLNTH, ZLNTH, NELZ
      READ(ICRD,103)    XINSRT, YINSRT, RADHL
      READ(ICRD,104)    IFLHL, ITYHL, LDMD, DSPLD
      CALL TITLE
      XLNTH2 = 2.D0 * XLNTH
      YLNTH2 = 2.D0 * YLNTH
      ZLNTH2 = 2.D0 * ZLNTH
      XINST2 = 2.D0 * XINSRT
      YINST2 = 2.D0 * YINSRT
      RADHL2 = 2.D0 * RADHL
      WRITE(IWRT,202)  XLNTH2,YLNTH2,ZLNTH2,NELZ, XINST2, YINST2, RADHL2
      IF(IFLHL .EQ. 1)    WRITE(IWRT,205)
      GO TO (91, 92, 93 ),  ITYHL
   91 WRITE(IWRT,207)
      GO TO 94
   92 WRITE(IWRT,208)
      GO TO 94
   93 WRITE(IWRT,209)
   94 IF(LDMD .EQ. 1)  GO TO 95
      WRITE(IWRT,203)  DSPLD
      GO TO 96
   95 WRITE(IWRT,204)  DSPLD
   96 CONTINUE
      READ(ICRD,100)   NMTL, NMLCH, (IMATL(J),IXMLCH(J),J=1,NMLCH)
      DO 71 IMTL=1,NMTL
      READ(ICRD ,1001)  MTLN, NTMP(IMTL), FIBORT(IMTL), ALFA1(IMTL),
     1        ALFA2(IMTL), ALFA3(IMTL)
      NTMP1 = NTMP(IMTL)
      DO 71 ITMP=1,NTMP1
   71 READ(ICRD ,1002)   TMPLL(IMTL,ITMP), (E(IMTL,J,ITMP),J=1,9)
      READ(ICRD,100)  NTYEL, NELCH,  (ITYEL(J),IXELCH(J),J=1,NELCH)
      CALL TITLE
      WRITE(IWRT,206)
      WRITE(IWRT,306) (IMATL(J), IXMLCH(J), J=1,NMLCH)
      I = 0
   97 CALL TITLE
      WRITE(IWRT,307)
   98 I = I+1
      IF(I .GT. NELCH)  GO TO 99
      WRITE(IWRT,306)    ITYEL(I),  IXELCH(I)
      LINE  =  LINE  +  1
      IF(LINE .LT. 48)  GO TO 98
      GO TO 97
```

```
 99 CONTINUE
    ZELTHS = ZLNTH/NELZ
    NGNP =136*(NELZ+1)
    NEL =20*NELZ
    NELZ1 = NELZ + 1
    DO 46 J=1,24
    IX(1,J) = IXDT1(J)
    IX(2,J) = IXDT2(J)
    IX(3,J) = IXDT3(J)
    IX(4,J) = IXDT4(J)
    IX(5,J) = IXDT5(J)
    IX(6,J) = IXDT6(J)
    IX(7,J) = IXDT7(J)
    IX(8,J) = IXDT8(J)
    IX(9,J) = IXDT9(J)
    IX(10,J) = IXDT10(J)
    IX(11,J) = IXDT11(J)
    IX(12,J) = IXDT12(J)
    IX(13,J) = IXDT13(J)
    IX(14,J) = IXDT14(J)
    IX(15,J) = IXDT15(J)
    IX(16,J) = IXDT16(J)
    IX(17,J) = IXDT17(J)
    IX(18,J) = IXDT18(J)
    IX(19,J) = IXDT19(J)
 46 IX(20,J) = IXDT20(J)
    IF(NELZ .EQ. 1 )  GO TO 47
    DO 48  IELZ=2,NELZ
    DO 48  I=1,20
    M =136*(IELZ-1)
    L =I+20*(IELZ-1)
    DO 48 J=1,24
 48 IX(L,J) = IX(I,J) + M
 47 CONTINUE
    IF(IFLHL .EQ. 0)  GO TO 75
    DO 78 IELZ=1,NELZ
    IELZM1 = IELZ-1
    L = 136 * IELZM1
    DO 83 I=1,24
    IX(NEL+1,I) = IXDTP1(I) + L
    IX(NEL+2,I) = IXDTP2(I) + L
    IX(NEL+3,I) = IXDTP3(I) + L
 83 IX(NEL+4,I) = IXDTP4(I) + L
    L =  9 * IELZM1
    DO 84 I=1,10
    IX(NEL+1,IDX1(I)) = NGNP + IPLNP1(I) + L
    IX(NEL+2,IDX2(I)) = NGNP + IPLNP2(I) + L
    IX(NEL+3,IDX3(I)) = NGNP + IPLNP3(I) + L
 84 IX(NEL+4,IDX4(I)) = NGNP + IPLNP4(I) + L
 78 NEL = NEL + 4
    NGNP = NGNP + 9*(NELZ+1)
```

```
75 CONTINUE                                                         DAH 2050
   IXELCH(NELCH+1) = 0                                              DAH 2060
   IXMLCH(NMLCH+1) = 0                                              DAH 2070
   I = 0                                                            DAH 2080
   J = 0                                                            DAH 2090
   DO 23 INEL=1,NEL                                                 DAH 2100
   IF(INEL .EQ. IXMLCH(I+1))   I=I+1                                DAH 2110
   IF(INEL .EQ. IXELCH(J+1))   J=J+1                                DAH 2120
   IX(INEL,25) = IMATL(I)                                           DAH 2130
   IX(INEL,26) = ITYEL(J)                                           DAH 2140
23 CONTINUE                                                         DAH 2150
   DO 33 I=1,NEL                                                    DAH 2160
33 IX(I,27) = ICLASS                                                DAH 2170
   DO 35 I=1,NCNP                                                   DAH 2180
35 MTLND(I) = 0                                                     DAH 2190
   X(1) = RADHL                                                     DAH 2200
   DO 49 I=7,13                                                     DAH 2210
49 X(I) = 0.D0                                                      DAH 2220
   X(14) =          (XINSRT-RADHL)/3.D0+RADHL                       DAH 2230
   X(16) = XINSRT / 3.D0                                            DAH 2240
   X(17) = X(16)                                                    DAH 2250
   X(18) =    2.D0 *  (XINSRT-RADHL)/3.D0+RADHL                     DAH 2260
   X(20) = 2.D0 *X(16)                                              DAH 2270
   X(21) = X(20)                                                    DAH 2280
   DO 50 I=22,28                                                    DAH 2290
50 X(I) = XINSRT                                                    DAH 2300
   X(29) =       (XLNTH-XINSRT) / 3.D0 + XINSRT                     DAH 2310
   X(30) = X(29)                                                    DAH 2320
   X(31) = X(29)                                                    DAH 2330
   X(32) = 2.D0*(XLNTH-XINSRT) / 3.D0 + XINSRT                      DAH 2340
   X(33) = X(32)                                                    DAH 2350
   X(34) = X(32)                                                    DAH 2360
   DO 51 I=35,41                                                    DAH 2370
51 X(I) = XLNTH                                                     DAH 2380
   Y(1) = 0.D0                                                      DAH 2390
   Y(7)=RADHL                                                       DAH 2400
   Y(8) =           (YINSRT-RADHL)/3.D0+RADHL                       DAH 2410
   Y(9) =      2.D0* (YINSRT-RADHL)/3.D0+RADHL                      DAH 2420
   Y(10) = YINSRT                                                   DAH 2430
   Y(12) = 2.D0 * (YLNTH-YINSRT)/3.D0 + YINSRT                      DAH 2440
   Y(13) = YLNTH                                                    DAH 2450
   Y(14) = 0.D0                                                     DAH 2460
   Y(11) =          (YLNTH-YINSRT)/3.D0 + YINSRT                    DAH 2470
   Y(16) = YINSRT                                                   DAH 2480
   Y(17) = YLNTH                                                    DAH 2490
   Y(18) = 0.D0                                                     DAH 2500
   Y(20) = YINSRT                                                   DAH 2510
   Y(21) = YLNTH                                                    DAH 2520
   Y(22) = 0.D0                                                     DAH 2530
   Y(23) = YINSRT / 3.D0                                            DAH 2540
   Y(24) = YINSRT / 3.D0 * 2.D0                                     DAH 2550
```

```
      Y(25) = YINSRT
      Y(26) = Y(11)
      Y(27) = Y(12)
      Y(28) = YLNTH
      Y(29) = 0.D0
      Y(30) = YINSRT
      Y(31) = YLNTH
      Y(32) = 0.D0
      Y(33) = YINSRT
      Y(34) = YLNTH
      DO 52 I=35,41
   52 Y(I) = Y(I-13)
      IF(IFLHL .EQ. 0)  GO TO 79
      NGNP1 =136*(NELZ+1)
      NEL1 =20*NELZ
      DO 80 IELZ=1,NELZ1
      M = NGNP1 + (IELZ-1)*9
      X(M+1)=-2.D0*RADHL/3.D0
      X(M+2) =      -RADHL/3.D0
      X(M+3) = 0.D0
      X(M+4) = -X(M+2)
      X(M+5) = -X(M+1)
      X(M+6) = 0.D0
      X(M+7) = 0.D0
      X(M+8) = 0.D0
      X(M+9) = 0.D0
      DO 86 I=1,5
   86 Y(M+I) = 0.D0
      Y(M+6) = X(M+5)
      Y(M+7) = X(M+4)
      Y(M+8) = X(M+2)
      Y(M+9) = X(M+1)
      DO 80 I=1,9
   80 Z(I+M) =(IELZ-1) * ZELTHS
   79 CONTINUE
      GO TO (1,2,3), ITYHL
    1 SXTHPI = 3.14159261358898D0  / 12.D0
      X(2) = RADHL  * DCOS(1.D0*SXTHPI)
      X(3) = RADHL  * DCOS(2.D0*SXTHPI)
      X(4) = RADHL  * DCOS(3.D0*SXTHPI)
      X(5) = RADHL  * DCOS(4.D0*SXTHPI)
      X(6) = RADHL  * DCOS(5.D0*SXTHPI)
      X(15) =        (XINSRT-RADHL*DCOS(3.D0*SXTHPI))/3.D0
     1       + RADHL*DCOS(3.D0*SXTHPI)
      X(19) =  2.D0*(XINSRT-RADHL*DCOS(3.D0*SXTHPI))/3.D0
     1       + RADHL*DCOS(3.D0*SXTHPI)
      Y(2) = RADHL*DSIN(1.D0*SXTHPI)
      Y(3) = RADHL*DSIN(2.D0*SXTHPI)
      Y(4) = RADHL*DSIN(3.D0*SXTHPI)
      Y(5) = RADHL*DSIN(4.D0*SXTHPI)
      Y(6) = RADHL*DSIN(5.D0*SXTHPI)
```

```
      Y(15) =      (YINSRT-RADHL*DSIN(3.D0*SXTHPI)) / 3.D0                  DAH 3070
     1       + RADHL * DSIN(3.D0*SXTHPI)                                     DAH 3080
      Y(19) = 2.D0*(YINSRT-RADHL*DSIN(3.D0*SXTHPI)) / 3.D0                  DAH 3090
     1       + RADHL * DSIN(3.D0*SXTHPI)                                     DAH 3100
      GO TO 7                                                                DAH 3110
    2 DO 87 I=2,4                                                            DAH 3120
      X(I) = RADHL                                                           DAH 3130
   87 Y(I+2) = RADHL                                                         DAH 3140
      X(5) = 2.D0 * RADHL/3.D0                                               DAH 3150
      X(6) =        RADHL/3.D0                                               DAH 3160
      Y(2) = X(6)                                                            DAH 3170
      Y(3) = X(5)                                                            DAH 3180
      X(15) = X(14)                                                          DAH 3190
      X(19) = X(16)                                                          DAH 3200
      Y(15) = Y(8)                                                           DAH 3210
      Y(19) = Y(9)                                                           DAH 3220
      GO TO 7                                                                DAH 3230
    3 X(2) = 5.D0 * RADHL/6.D0                                               DAH 3240
      X(3) = 2.D0 * RADHL/3.D0                                               DAH 3250
      X(4) =        RADHL/2.D0                                               DAH 3260
      X(5) = X(3) / 2.D0                                                     DAH 3270
      X(6) = X(5) / 2.D0                                                     DAH 3280
      DO 88 I=2,6                                                            DAH 3290
   88 Y(I) = X(8-I)                                                          DAH 3300
      X(15) = (X(14) + X(16))/2.D0                                           DAH 3310
      X(19) = (X(18) + X(20))/2.D0                                           DAH 3320
      Y(15) = (Y( 8) + Y(23))/2.D0                                           DAH 3330
      Y(19) = (Y( 9) + Y(24))/2.D0                                           DAH 3340
    7 CONTINUE                                                               DAH 3350
      DO 601 I=1,6                                                           DAH 3360
      X(41+I) = -X(I)                                                        DAH 3370
  601 Y(41+I) =  Y(I)                                                        DAH 3380
      DO 602 I=14,41                                                         DAH 3390
      X(34+I) = -X(I)                                                        DAH 3400
  602 Y(34+I) =  Y(I)                                                        DAH 3410
      DO 603 I=2,13                                                          DAH 3420
      X(74+I) = -X(I)                                                        DAH 3430
  603 Y(74+I) = -Y(I)                                                        DAH 3440
      DO 604 I=15,17                                                         DAH 3450
      X(73+I) = -X(I)                                                        DAH 3460
  604 Y(73+I) = -Y(I)                                                        DAH 3470
      DO 605 I=19,21                                                         DAH 3480
      X(72+I) = -X(I)                                                        DAH 3490
  605 Y(72+I) = -Y(I)                                                        DAH 3500
      DO 606 I=23,28                                                         DAH 3510
      X(71+I) = -X(I)                                                        DAH 3520
  606 Y(71+I) = -Y(I)                                                        DAH 3530
      X(100) = -X(30)                                                        DAH 3540
      X(101) = -X(31)                                                        DAH 3550
      X(102) = -X(33)                                                        DAH 3560
      X(103) = -X(34)                                                        DAH 3570
```

```
      Y(100) = -Y(30)
      Y(101) = -Y(31)
      Y(102) = -Y(33)
      Y(103) = -Y(34)
      DO 607 I=36,41
      X(68+I) = -X(I)
  607 Y(68+I) = -Y(I)
      DO 608 I=2,6
      X(108+I) = X(I)
  608 Y(108+I) =-Y(I)
      DO 609 I=88,109
      X(27+I) =-X(I)
  609 Y(27+I) = Y(I)
      DO 53 I=1,136
   53 Z(I) = 0.D0
      DO 54 IELZ=1,NELZ
      XL = IELZ * ZELTHS
      DO 54 I=1,136
      L = IELZ * 136 + I
      Z(L) = XL
      Y(L) = Y(I)
   54 X(L) = X(I)
      DO 55 I=1,NGNP
      ICODE(I) = 0
      UX(I)=0.D0
      UY(I)=0.D0
   55 UZ(I)=0.D0
      DO 58 I=1,136
   58 ICODE(I)     =5
      L = 136*(NELZ+1)
      DO 57 I=1,9
   57 ICODE(I+L) = 5
      IF(LDMD .EQ. 1)  GO TO 60
      DO 56 I=36,41
      ICODE(I    ) = 3
      ICODE(I+34) = 3
      ICODE(I+68) = 3
      ICODE(I+95) = 3
      UX(I    ) = DSPLD
      UX(I+95) = DSPLD
      UX(I+68) =-DSPLD
      UX(I+34) =-DSPLD
      DO 56 IELZ=1,NELZ
      L = 136 * IELZ
      ICODE(I+L    ) = 1
      ICODE(I+L+34) = 1
      ICODE(I+L+68) = 1
      ICODE(I+L+95) = 1
      UX(I+L    ) = DSPLD
      UX(I+L+95) = DSPLD
      UX(I+L+34) =-DSPLD
```

```
56 UX(I+L+68) =-DSPLD
   ICODE(35) = 7
   ICODE(69) = 7
   UX(35) = DSPLD
   UX(69) =-DSPLD
   DO 63 IELZ=1,NELZ
   L = 136*IELZ
   ICODE(35+L) = 1
   ICODE(69+L) = 1
   UX(35+L) = DSPLD
63 UX(69+L) =-DSPLD
   GO TO 9
60 CONTINUE
   IF(NELZ .EQ. 1)   GO TO 65
   DO 64 IELZ=2,NELZ
   M =136*(IELZ-1)
   DO 72 I=35,38
72 UX(I+M) = XLDSQ(I-34) * YINSRT*ZELTHS * DSPLD/8.D0
   DO 73 I=38,41
73 UX(I+M) = UX(I+M) +  (XLDSQ(I-37) * (YLNTH-YINSRT)*ZELTHS
  1                    *DSPLD/8.D0 )
64 CONTINUE
65 DO 66 I=35,38
   UX(I)    = XLDSQ(I-34) * YINSRT*ZELTHS * DSPLD/16.D0
   M =I+136*NELZ
66 UX(M)    = XLDSQ(I-34) * YINSRT*ZELTHS * DSPLD/16.D0
   DO 74 I=38,41
   M =I+136*NELZ
   UX(M)    = UX(M)    +  (XLDSQ(I-37) * (YLNTH-YINSRT)*ZELTHS
  1                    *DSPLD/16.D0)
74 UX(I)    = UX(I)    +  (XLDSQ(I-37) * (YLNTH-YINSRT)*ZELTHS
  1                    *DSPLD/16.D0)
   NELZP1 = NELZ+1
   DO 81 I=36,41
   DO 81 IELZ=1,NELZP1
   M = 136 * (IELZ-1) + I
   UX(M+34) = -UX(I)
   UX(M+68) = -UX(I)
81 UX(M+95) =  UX(I)
   DO 82 I=1,NELZP1
   M = 136*(I-1)
   UX(M+35) = 2.D0 * UX(M+35)
82 UX(M+69) = -UX(M+35)
   ICODE(35) = 6
   ICODE(69) = 6
   ICODE(13) = 3
   ICODE(87) = 3
 9 RETURN
   END
```

```
      SUBROUTINE TITLE
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
C   *  SUBROUTINE TITLE PRINTS THE HEADING ON EACH PAGE                 *
C   *                                                                   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE
  100 FORMAT (1H1,'FEM 72-DOF GENERAL HEXAHEDRONS THERMO-ELASTIC, VARYIN
     1G MATERIAL PROPERTIES, DANA', 9X, 'PAGE', I3)
  101 FORMAT (1H0,10A8 )
      WRITE (IWRT,100) IPAGE
      WRITE (IWRT,101) HED
      IPAGE= IPAGE +1
      LINE = 0
      RETURN
      END
```

```
      SUBROUTINE MODF                                                   MOH
C                                                                       MOH
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MOH
C     *                                                               * MOH
C     *  SUBROUTINE MODF DOES NOTHING -- THIS SUBROUTINE CAN BE USED TO * MOH
C     *  MODIFY ANY INFORMATION THAT HAS BEEN GENERATED                * MOH
C     *                                                               * MOH
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MOH
C                                                                       MOH
      IMPLICIT REAL*8 (A-H,O-Z)                                         MOH
      INTEGER*2   IX, ICODE, IDPIX1, MTLND                              MOH
      COMMON /GENMAT/ X(1015),Y(1015),Z(1015),UX(1015),UY(1015),UZ(1015)MOH
     1 ,  TMPND(1015), BCTMP,                                           MOH
     2 ALFA1(10),ALFA2(10),ALFA3(10),FIBORT(10),E(10,9,10),TMPEL(10,10),MOH
     3  NTMP(10),IX(144,27),ICODE(1015),IDPIX1(19,19,11),MTLND(1015)    MOH
      COMMON  / GENL1 /  TMPINT, EPSTMP, AX, BY, CZ, THELLX,THELLY,ELLZ,MOH
     1    NEL, NGNP, NGLDF, NMTL, NTYEL, LMTMP, NELX, NELY, NELZ, ICLASSMOH
     2   , ITYTD, NELX31, NELY31, NELZ1                                 MOH
      COMMON / HEAD /  HED(10),ICRD,IWRT,IPAGE,LINE                     MOH
      RETURN                                                            MOH
      END                                                               MOH
```